



Address comments to:
Venue
User Documentation
1549 Industrial Road
San Carlos, CA 94070
415-508-9672

NOTECARDS USER'S GUIDE

Release 2.0

August, 1991

Copyright © 1990, 1991 by Venue.

All rights reserved.

Medley is a trademark of Venue.

Xerox® is a registered trademark and InterPress is a trademark of Xerox Corporation.

UNIX® is a registered trademark of UNIX System Laboratories.

PostScript is a registered trademark of Adobe Systems Inc.

The X Window System is a trademark of the Massachusetts Institute of Technology.

IBM, AIX and RISC SYSTEM/6000 are trademarks of International Business Machines, Inc.

MIPS is a registered trademark of MIPS Computer Systems, Inc.

Copyright protection includes material generated from the software programs displayed on the screen, such as icons, screen display looks, and the like.

The information in this document is subject to change without notice and should not be construed as a commitment by Venue. While every effort has been made to ensure the accuracy of this document, Venue assumes no responsibility for any errors that may appear.

Text was written and produced with Venue text formatting tools; Xerox printers were used to produce text masters. The typeface is Classic.

PREFACE

Welcome to NoteCards, an idea manipulation tool for the modern intellectual. NoteCards provides you with a new way to organize and manage your ideas on your computer desk top.

This guide explains in detail how to install NoteCards on your Sun 3 or Sun 4 Workstation, provides a basic tutorial for the NoteCards environment, and contains the reference manual for NoteCards.

Audience

The *NoteCards Users's Guide* was written for users of NoteCards on Sun Workstations. The *Guide* assumes that you are already familiar with UNIX and SunOS concepts. This *Guide* is divided into four parts:

Part I, Introduction and Installation, should be read by the System Administrator or person installing the NoteCards system. The NoteCards user should also read this section, in particular, Chapters 1 and 4, Introduction and System Use Issues. Chapters 2 and 3, System Requirements and Software Installation, should be read by those interested in the system-level aspects of UNIX and NoteCards. (The system installer should also read Appendices D and E.)

Part II, NoteCards Tutorial, provides an introduction to NoteCards and the system which underlies it. Chapter 5, NoteCards Basics, is the place for all new users of NoteCards to start once they have their system up and running.

Part III, Reference Manual, is a compilation of all the necessary background information on NoteCards new and old users will need to learn to make full use of the NoteCards system.

The final part contains the Appendices, Glossary, and Index.

Included with this *Guide* are the manuals for TEdit and Sketch, *A User's Guide to TEdit* and *A User's Guide to Sketch*.

What's in the Manual

Here is what you will find in this manual, chapter by chapter, accompanied by a brief description of when you will want to read each chapter.

Chapter 1, Introduction, gives a brief introduction to and some background on NoteCards as well as explaining the stylistic conventions used in the *Guide*. Read this chapter before you install NoteCards.

Chapter 2, System Requirements, describes the hardware requirements of the system. Read this chapter before you install NoteCards on a particular machine to verify that it will run acceptably on that machine.

Chapter 3, Software Installation, gives you step-by-step directions for installing NoteCards. Read this chapter before and during system installation.

Chapter 4, System Use Issues, discusses some of the basic ins and outs of using the system, initialization, starting, and exiting NoteCards, as well as file access. Read this

chapter after you have installed NoteCards. In particular, read the Starting NoteCards and Exiting NoteCards and Saving State sections when you are modifying your .cshrc and .login UNIX files for ideas on how to simplify the NoteCards booting process. To allow it to run on two different file systems, NoteCards differs slightly from the standard SunOS and UNIX file systems. Read the Using SunOS Files from NoteCards section when you are learning how to access files from NoteCards.

Chapter 5, NoteCards Basics, leads you step by step through the basics of using the NoteCards system. This is the place for new users to start once NoteCards is up and running. This is also a good review chapter as it brings together ideas scattered throughout the user documentation.

Chapter 6, Building NoteCards Structures, gives you some insights on how to use the NoteCards sytems to its best advantage. The information is based on user experience with NoteCards over several years. Read this chapter after you have become somewhat familiar with NoteCards.

Chapter 7, User Interface, covers some of the more basic aspects of using the system—the mouse, keyboard, menus, etc. Read this chapter to understand more of the lower level capabilities of the system.

Chapter 8, Links, describes the link icon in all of its manifestations. Links and their physical representation, link icons, live at the core of NoteCards. Read and understand this chapter as soon as possible to make the most effective use of the system.

Chapter 9, Cards and Banners, discusses the user interface to individual notefiles and those aspects that all cards have in common. This is also an important chapter. The Card Menu section is particularly important as it explains the functionality all cards have in common.

Chapter 10, User Cards, discusses cards where you are responsible for creating the contents. Read the Text Cards and FileBox Cards sections right away. The Text Cards section points you to the TEdit manual, *A User's Guide to TEdit*. As TEdit forms the core of the system, we suggest that you make an effort to gradually learn more about TEdit all the time. Read the remaining sections as you explore the other card types. Note that the Bit Map Editor is discussed at the end of this chapter even though there is no bit-map card type.

Chapter 11, System Cards, covers those cards where the system creates the contents for you. The Browser Cards section is the most important one to read for a beginner. The others can be read as you need them.

Chapter 12, MenuBox Icon, discusses the options on the three menus available from the MenuBox. The important commands for a new users are on the Notefile Ops menu. When just getting used to NoteCards, focus on the Open, Checkpoint, Close, and Abort commands.

Chapter 13, System Parameters, explains how to change the global defaults for the system. Since most users will find that the default settings will suit all their needs, new users may want to skip this chapter.

Chapter 14, FileBrowser, explains how to use the FileBrowser.

Chapter 15, Other Tools, explains how to use the Directory Connector, and the clocks.

Chapter 16, Printing, covers setting up print capabilities and printing to PostScript and Interpress printers. Read this chapter when you have the machine on a network and are ready to set up your print capability.

Chapter 17, Known Problems, Error Conditions and Recovery, covers known problems and how to cope with and avoid them. A quick reading of the Known Problems and Break Windows sections are all that most users will ever need.

Appendix A, Notefile Concepts, explains the inner workings and structure of notefiles. Read this if you want to understand how information is stored in the notefile and why you have to perform operations like compacting occasionally. This is essential background material if you are about to inspect or fix a damaged notefile.

Appendix B, Programmer's Interface, explains to a programmer how to obtain software interface to NoteCards in Interlisp.

Appendix C, Notefile Inspector, explains how to fix notefiles damaged by power failed, broken net connections and other unforeseeable calamities. Read this before you attempt to fix a damaged notefile.

Appendix D, Initialization Files, collects a lot of information on how to write a file which will automatically set site specific values for parameters like printer names and font locations. Read this appendix when you are first setting up your system.

Appendix E, Checksum Control, covers what to do if you believe the files loaded from the distribution tape are damaged.

Acknowledgments

NoteCards has a long history. Its gestation began in October 1982 when the U.S. Government funded XSYS (Xerox Special Information Systems) and Xerox PARC (Palo Alto Research Center) to prototype a Problem-Structuring-Aids system to allow users to build "semantic networks of textual information." The first successful prototype was completed by Frank Halasz, at PARC, in November 1983. The first real user of the NoteCards system was Ken Allen, a history graduate student at Stanford University, in the spring of 1984. The birth came in June 1984 when NoteCards was released to the government, in fulfillment of the Problem-Structuring-Aids contract. Since then, XSYS and PARC have released three unsupported versions of NoteCards on Xerox D-machines. In November 1988, the NoteCards product team started work on a supported, productized version of NoteCards. NoteCards Release 2.0 for the Sun Workstation was completed in August 1991.

NoteCards is based on the work of many people.

From Xerox, Frank Halasz, Tomas Moran, Randall Trigg, Richard Burton, Ronald Kaplan, Peggy Irish, Catherine Marshall, and many others.

From Envos, Robert Krivacic, Keith Mountford, Craig Sweat, Karin Sye, Daniel Sagalowicz, Larry Harada, John Sybalsky, and others.

[This page intentionally left blank]

TABLE of CONTENTS

Preface	xvii
----------------------	------

Part I - Introduction and Installation

1. Introduction	1
Operating System Conventions	1
System Components	1
NoteCards Device-Naming Conventions	2
Notation Conventions	2
Font	2
Prompts	3
Keyboard Conventions	3
2. System Requirements	5
Prerequisites	5
Processor Hardware	5
Memory	5
Swap Space	5
Disk Space	5
Input/Output Devices	5
Bitmap Display	5
Printers	6
Tape Access	6
Operating System Requirements	6
Constraints	6
Release Contents	6
Documentation	6
Software	7
3. Software Installation	9
Ensuring Adequate Swap Space	9
Installing Software	9
Copy Protection	10
Configuring the Software	11
Relinking	11
Enabling PUP/XNS Ethernet	11
Using/Installing the Host Access Key	12

4. System Use Issues	13
Site Initialization File	13
Starting NoteCards	13
Where NoteCards Looks for Your Sysout.....	14
Exiting NoteCards and Saving State	14
Keyboard Interpretation	15
Sun Type 3 Keyboard	16
Sun Type 4 Keyboard	17
Console Messages	18
File Compatibility	18
Sysout Compatibility between Sun and Xerox Workstations	18
File Compatibility between Sun and Xerox Workstations.....	18
Using SunOS Files from NoteCards	19
File Naming Conventions	19
Common {DSK} and {UNIX} Naming Conventions	20
{DSK} Naming Conventions	20
Version Numbering.....	21
Pathnames	22
{UNIX} Naming Conventions.....	22
Directories	23
Directory Enumeration.....	23
Directory Creation.....	24
Directory Delection	24
Open File Limit.....	24
Default Pathname	24
File Attributes	25
File System Errors	25

Part II - NoteCards Tutorial

5. NoteCards Basics	27
A Few Pointers on Mouse Etiquette	27
Mouse Button Use.....	27
Left Mouse Button	28
Middle Mouse Button.....	28
Right Mouse Button.....	28

Regions Sensitive to Mouse Activity	28
Background	28
Windows and the Window Title Bar	29
Icons	31
Scroll Bars	31
Dealing with Simple Error Conditions	32
Starting NoteCards	32
Opening a Notefile	32
Creating a New Notefile	34
Closing Notefiles and Exiting NoteCards	35
Closing Notefiles	35
Close	35
Abort	35
Existing Notefiles.....	36
Using NoteCards	36
Opening Cards / Traversing Links.....	37
Shrinking Cards.....	38
Closing Cards.....	39
Creating Cards	39
Where to Go From Here	39

6. Building NoteCards Structures	41
Introduction	41
A Standard View of Building NoteCards Structures.....	41
A Non-Standard View of Building NoteCards Structures.....	41
FileBoxes and Hierarchies	41
Flat System.....	42
Almost Flat System.....	42
Link Type.....	42
Multiple Users	42
Cards and Information Chunk Size	42
Types of Structures Built with Notecards.....	42
Specializing NoteCard	43

Part III - Reference Manual

7. User Interface	45
Window Menu	45
Icon Menu	52
Background Menu.....	53

8. Links	61
Links and Link Types	61
System-Reserved Link Types	61
User-Specified Link Types	62
Link Directions and Link Ends	63
Link Categories	63
Other Link Terminology	63
Link Types vs. Card Types	63
Link Access and Use	64
Link Icon Functionality	64
Active Regions of Link Icons	64
Mouse-Button Actions in Link Icons	65
Viewing Local and Global Links	67
Unfiled and Lost Cards	68
Creating Links	69
Creating Links in Text Cards	69
Creating Links in Other Card Types	72
Text-Based Cards	72
Sketch-Based Cards	72
Graph-Based Cards	73
Deleting Links	73
Deleting Links from Card Contents	73
Text-Based Cards	73
Sketch-Based Cards	74
Graph-Based Cards	74
Deleting Links from the Show Links Display	74
Tailoring Links	75
Link Ops Menu	75
System Parameters for Links	76
Cross-File Links	77
Appearance	77
Missing Notefiles	77
9. Cards and Banners	79
Notefile Banner	79
Banner Title Bar	79
Notefile Ops Menus	79
Middle-Mouse-Button Title-Bar Menu	81
Full File Name	81
File Capacity	81

Special Cards	82
Table of Contents FileBox.....	83
To Be Filed FileBox.....	83
Orphans FileBox	83
New Cards	83
User Cards and System Cards	84
Text-, Sketch-, and Graph-Based Cards	85
Card Menu	85
Standard Card Menu.....	85
Edit Property List	85
Show Links	86
Show Info.....	86
Designate FileBoxes.....	87
Assign Title	88
Title/FileBoxes	88
Insert Link	88
Close and Save	89
FileBox Card Menu	90
10. User Cards	91
Text Cards	91
Text-Card Menu.....	91
FileBox Cards	92
Suggested FileBoxes and Note Cards	92
FileBox-Card Menu	93
System Parameters Affecting FileBoxes.....	93
Sketch Cards	93
Sketch-Card Menu	93
System Parameters Affecting Sketch Cards	94
Graph Cards	94
Graph-Card Menu	94
Grapher Menu	94
Graph-Card Options (and Grapher Menu Options)	95
Move Node (Move Node)	95
Remove Node (Delete Node)	96
Connect Nodes (Add Link)	96
Disconnect Nodes (Delete Link).....	97
Add Label (Add Node).....	97
Change Label (Change Label)	97
Smaller Label (Label Smaller)	97
Larger Label (Label Larger).....	97

Toggle Shade (<-> Shade)	97
Toggle Border (<-> Border)	97
Directed/Undirected (<-> Directed).....	98
Sides/Centers (<-> Sides)	99
Fix Menu.....	99
(Stop).....	99
Bitmap Editor	99
Inserting Bitmaps into Cards	99
Text-Based Cards.....	99
Sketch-Based Cards.....	100
Bitmap Operations	100
Change Scale	101
Hand Edit.....	101
Trim	101
Reflect Left-to-Right.....	101
Reflect Top-to-Bottom	101
Reflect Diagonally.....	101
Rotate Left	101
Rotate Right.....	101
Expand on Right.....	102
Expand on Left	102
Expand on Bottom	102
Expand on Top	102
Switch Black & White.....	102
Add Border	102
Bitmap Editor.....	103
Paint	104
ShowAsTile.....	105
Grid On/Off	105
GridSize.....	105
Reset	105
Clear	105
Cursor.....	106
OK	106
Abort	106
11. System Cards	107
Search Cards.....	107
LinkIndex Cards	108

Document Cards	109
HeadingsFromFileBoxes	110
TitlesFromNoteCards	110
BuildBackLinks	111
CopyEmbeddedLinks	111
ExpandEmbeddedLinks	111
Browser Cards	112
Recompute Browser	113
Multiple Roots	114
Dashed Links	114
Arrowheads	114
Browser Specs	114
Middle-Button Title-Bar Menu Options	115
Editing the Browser Manually and Structure Editing	116
Overview Windows on Browser Cards	118
Creating an Overview Window	118
Reshaping the Overview Window	118
Scrolling and the Wire Frame	118
Recomputing the Overview Contents	119
Browser Overview Stylesheet	119
Tidbits	119
12. MenuBox Icon	121
Notefile Options	121
Open	122
Compact	123
Inspect & Repair	124
Copy	124
Rename	124
Delete	124
Create	125
Checkpoint	125
Close	127
Abort	127
NC FileBrowser	128
Card Options	128
Close	129
Delete	130
Copy	130
Move	131

Other Options	131
Edit Parameters	131
NF Indicators On	131
TEdit Killer On	132
Load Patches	133
13. System Parameters	135
Accessing System Parameters	135
NoteCards System Parameters	136
Extra TEdit Props	136
Include Card Object In ShowInfo	137
Del TEdit Process When Shrinking	137
Font Parameters	137
Menu Font	138
Default Font.....	138
Link Icon Font.....	138
Notefile Parameters	139
Show Notefile On Cards.....	139
New Notefile Initial Size	140
Menu Lingers After Notefile Close	140
Card Parameters	140
Force Filing	140
Force Titles	140
Default Card Type	140
Close Cards Off Screen	140
Bring Up Cards At Previous Pos	141
FileBox Card Parameters	141
Markers In FileBoxes	141
Alphabetize FileBox Children	141
Browser Card Parameters	141
Special Browser Specs	141
Arrow Heads In Browsers	142
Link Dashing In Browsers	142
Sketch Card Parameters	142
Attach Sketch Menu	142
Link Parameters.....	142
Link Icon Border Width.....	142
Link Icon Multi Line Mode	143
Cross File Link Mode	143
Link Icon Max Width in Pixels	143
Link Icon Show Title Default.....	144

Link Icon Attach Bitmap Default	144
Link Icon Show Link Type Default.....	144
Use Deleted Link Icon Indicators	144
14. FileBrowser	145
User Interface	145
Starting FileBrowser.....	145
Specifying What Files to Browse.....	145
Examples.....	146
Using the FileBrowser Window	146
Selecting Files.....	148
Commands that Require Input	149
Aborting Commands	149
Quitting the FileBrowser	150
Copy-Selecting Files	151
Getting Hardcopy Directory Listings.....	151
FileBrowser Commands	151
Delete, Undelete	151
Copy.....	153
Rename.....	154
Hardcopy.....	154
See	155
FileBrowse	157
Edit.....	157
Load	158
Compile	158
Expunge.....	158
Recompute.....	158
New Info.....	160
Set Depth	161
Sort.....	162
Troubleshooting Problems with FileBrowser	162
15. Other Tools	163
Digital Clock.....	163
Starting and Stopping the Digital Clock.....	163
Changing the Digital Clock.....	164
Set Font	164
Set Time	165
Set Alarm.....	165
Quiet Alarm/Loud Alarm	166

Delete Alarm Setting.....	166
Shape to Fit	166
12-Hour/24-Hour Clock.....	166
Set Local Time Zone	166
Add New Regional Time Zone.....	166
Delete This Window.....	166
Set Font for Aux Clocks.....	167
Set Aux Clock Font In Just This Window.....	167
Set Time-Zone Heading	167
Set Regional Time Zone.....	167
Analog Clock.....	167
Starting the Analog Clock.....	168
Stopping the Analog Clock.....	168
Changing the Analog Clock.....	168
Numbers	169
Points.....	169
No Numbers	169
Rings	169
No Rings.....	169
Hands	169
No Hands.....	169
Times.....	169
No Times	169
Show Style.....	169
Set to Default.....	170
Change Default.....	170
Set Time	170
Known Problems	170
Directory Connector.....	170
Starting the Directory Connector	170
Stopping the Directory Connector	170
Changing the Directory Connector Fonts.....	170
Using the Directory Connector	171
Left Mouse Button	171
Middle Mouse Button.....	171
16. Printing	173
Setting up Printers	173
Adding Default Printers.....	173
Setting the Default Printer	174
Removing a Default Printer	175

Setting Font Directories	175
Adding a New Font Directory	175
Showing the Font Directories	176
Removing a Font Directory	177
Troubleshooting Common Printer Problems.....	178
Can't Determin IMAGETYPE for {LPT}	178
Font not Found.....	179
Can't Convert a POSTSCRIPT for a NIL Printer	179
Customizing Your PostScript Printer	179
Portrait vs. Landscape	179
Maximum Wild Font Size	180
Font Association Menu.....	180
PostScript Short Edge Shift.....	181
PostScript Short Edge Points.....	181
PostScript Long Edge Shift	182
PostScript Long Edge Points	182
Landscape Mode.....	182
Landscape Text Mode.....	182
Bitmap Scape.....	183
Texture Scale	183
Image Size Factor	183
17. Error Recovery and Known Problems	185
System Status, Aborting Operations, and Spawning a New Mouse	185
System Status	185
Aborting Operations	185
Spawning a New Mouse	186
Break Windows.....	186
Lisp Errors	187
Errors While Running NoteCards.....	187
Xerox Workstation-Specific Errors	187
Virtual Memory Errors	188
URAIID	188
Entering URAID.....	188
URAIID Commands.....	188
Other Fatal Error Conditions.....	189
System Error Conditions	189

Known Problems.....	191
Printing Browsers	191
Notefile Indicator Window	191
Fonts	191
Tedit	191

Appendices, Glossary, and Index

Appendix A. Notefile Concepts.....	193
Background Concepts	193
Unique Identifiers: UIDs	193
Card Parts	193
Notefile Structure: Index and Data Areas	193
Notefile Checkpointing	195
Storing Links and Repairing Notefiles with Inconsistent Links.....	196
Appendix B. NoteCards Programmer's Interface.....	197
Notefile Creation and Access	197
Creating and Accessing NoteCard Types.....	199
Creating Notecards and Fileboxes	203
Accessing Notecards and Fileboxes.....	206
Creating and Accessing Links	211
Creating and Accessing Link Labels.....	213
Customizing the NoteCards Interface.....	214
Handy Miscellaneous Functions	217
Appendix C. Notefile Inspector.....	225
Running the Notefile Inspector: Phase 1: Scouring the Data Area.....	225
Continue Repair	226
End Inspect&Repair	226
Abort.....	227
Recheck Bad Cards	227
Inspect Cards	227
Running the Notefile Inspector: Phase 2: Your Chance to Tinker	227
Card Inspector Menu.....	228
Card Parts Inspector	229
Cancel	231
Update.....	231
Delete.....	231
Undelete.....	231
Reset.....	231
Running the Notefile Inspector: Phase 3: Rebuilding Your Links	232

Tips and Hints for Using Inspect&Repair	233
When in Doubt, Abort!	233
Fixing versus Tinkering	233
Inspect&Repair After a Crash	233
Compacting	233
Inspect&Repair Closes the Notefile Before Starting	233
Fixing Enough Problems to Allow Phase 3 to Run	234
Appendix D. Initialization Files	235
Size Initialization File	235
Locating the Initialization File	235
Initialization File Structure	236
Site Variables	236
Appendix E. Checksum Control	241
Commands	241
Output	241
Examples	242
Glossary	GLOSSARY-1
Index	INDEX-1

[This page intentionally left blank]

1. INTRODUCTION

NoteCards is a hypertext system that helps you formulate, structure, manage, and present ideas. WYSIWYG text and graphic editors let you create semantic networks of electronic notecards connected by typed links. NoteCards provides specialized tools for manipulating and navigating through these networks.

NoteCards is a computer environment designed to help people work with ideas. Its users are authors, researchers, designers, and other intellectual laborers engaged in analyzing information, constructing models, formulating arguments, designing artifacts, and generally processing ideas. The system provides these users with a variety of tools for collecting, representing, managing, interrelating, and communicating ideas. NoteCards is based on the notion that creative intellectual work is a hand-craft, a uniquely human skill that cannot be easily automated.

NoteCards provides the user with a "semantic network" of electronic notecards interconnected by typed links. This network serves as a medium in which the user can represent collections of related ideas. It also functions as a structure for organizing, storing, and retrieving information. The system provides the user with tools for displaying, modifying, manipulating, and navigating through this network.

From: *NoteCards in a Nutshell*, by Frank G. Halasz, Thomas P. Moran, Randall H. Trigg of the Intelligent System Laboratory in the Xerox Palo Alto Research Center, Proceedings of the ACM CHI+GI '87 Conference, Toronto, Canada April 1987

Operating System Conventions

SunOS is Sun Microsystems' version of the UNIX operating system. AIX is IBM's version of the UNIX operating system. For users unfamiliar with the Sun Workstation or RS/6000, the following operating system conventions, which differ from corresponding Lisp conventions, are used in the manual.

For complete information on UNIX and your operating system, refer to the documentation set provided with your operating system.

case, filenames	Type-in to UNIX is case sensitive. Typically, input is in lowercase letters. When UNIX searches for a name, it is case sensitive; it distinguishes between lower- and uppercase characters. By convention, most names are in lowercase characters.
shell	Command interpreter. The commands shown are in the C-Shell, unless otherwise noted.

System Components

Functionally, the NoteCards system consists of the following components:

emulator	A UNIX-executable program, which performs sexecutes the NoteCards system contained in the sysout and provides access to the host's hardware.
----------	--

sysout	A virtual memory image (the sysout) containing both the NoteCards program and data structures. The sysout provided can be used both on the RS/6000 and Sun Workstations.
notefiles	A file containing your cards on a particular topic and all their links. This is your data and its organizational structure.
fonts	Data describing the "looks" of printed characters used by NoteCards' graphics, windowing, and hardcopying subsystems. Font directories are in four groups: display fonts, PostScript printer fonts, InterPress printer fonts, and Press printer fonts.

NoteCards Device-Naming Conventions

NoteCards lets you interact with UNIX file systems (including file systems mounted from other machines) by using host device names. The two device names are as follows:

- {DSK} A host name which gives you access to the UNIX file system using Xerox workstation local disk conventions.
- {UNIX} A host name which gives you access to the file system using normal UNIX conventions.

The {DSK} device name provides an interface to the Sun Workstation and RS/6000 for users who want to maintain compatibility with existing development tools and applications originally developed on a Xerox workstation. The {UNIX} device name provides a way for new applications to interact naturally with UNIX. Chapter 5 explains, in greater detail, some important exceptions and restrictions to the {DSK} and {UNIX} device name.

Notation Conventions

Text marked by a revision bar in the right margin contains information that was added or modified since the last release. Fonts, packages, and prompts have the following types of notation.

Fonts

Bold text in TITAN font indicates text you should type in exactly as printed.

Regular TITAN font text indicates what the system prints on your workstation screen. Lisp functions and variables and UNIX files and programs are also shown in TITAN FONT.

Text in Bold Classic indicates menu and option names, menu commands, and NoteCards parameters.

Text in Classic italics indicates variables or parameters that you should replace with the appropriate word or string.

Prompts

All examples which include UNIX dialogues use the following conventions for the UNIX prompt.

A number sign (#), part of the system prompt, indicates that you are logged on as `root` or are running `su`; for example,

```
prompt#
```

A percent sign (%), part of the system prompt, indicates that a user other than `root` is logged on; for example,

```
prompt%
```

Keyboard Conventions

Keys that you press are in uppercase (e.g., `COPY`, for the Copy key). A carriage return is indicated as `RETURN`. Instructions that ask you to press two or three keys simultaneously are indicated as follows:

Press `CONTROL-E`

[This page intentionally left blank]

2. SYSTEM REQUIREMENTS

This chapter outlines the hardware and system requirements for running NoteCards on a Sun Workstation. It also describes the contents of the software release and documentation provided.

Prerequisites

Processor Hardware

NoteCards runs on Sun 3 and Sun 4 Workstations. It runs on both standalone workstations and diskless workstations linked to servers.

NoteCards on the Sun 3 Workstation requires the MC68881 floating point coprocessor chip; all Sun 3 Workstations are currently sold with this chip. On the Sun 4 Workstation, the Weitek 1164/1165 coprocessor is optional.

For adequate performance, we recommend at least a 20 MHz 68020 (Sun 3/60 or 3/260), or a 14 MHz SPARC (Sun 4/110 or 4/260), or a SPARCstation.

Memory

You can expect reasonable interactive performance with 8 megabytes (Mb) or more of RAM. Smaller configurations of diskless workstations have been tested, but performance suffers.

Swap Space

NoteCards requires 45Mb of swap space for its own use. NoteCards reserves this space at startup, but its requirement does not grow.

Disk Space

You need a minimum of 17 Mb of disk file space for loading the software from tape, and an additional 1 Mb of disk file space (on the file system where the installation is taking place) to install and configure the NoteCards image. The 17 Mb of disk space needed for the NoteCards software can be broken down as follows:

NoteCards sysout	7 Mb
Fonts	7 Mb
Byte code emulator	3 Mb

Input/Output Devices

NoteCards gives you access to the Sun's input/output devices, such as display, keyboard, mouse, and file systems. It also provides access to PUP and XNS Ethernet services directly.

Bitmap Display

NoteCards only runs on machines with monochrome displays or with color displays that can be operated in single-bit-per-pixel mode. NoteCards supports both the standard resolution display (1152 x 900) and the high-resolution display (1600 x 1280).

Printers

For hardcopy output, NoteCards allows you to print to a Postscript printer, via NFS, or to a Xerox Interpress or Press printer, via PUP/XNS.

Tape Access

For installation you need a ¼-inch cartridge tape, located locally or on a remote machine.

Operating System Requirements

NoteCards on the Sun 3 workstation requires SunOS operating system versions 3.2, 3.4, 3.5, 4.0, 4.0.3, or 4.1. On the Sun 4 workstation, NoteCards requires SunOS version 4.0, 4.0.3, or 4.1.

Note: Notecards XNS Ethernet code cannot be run simultaneously with SunOS 3.5 Kernal XNS Ethernet code.

Constraints

When NoteCards is running, it takes over the entire display screen. Other window systems such as suntools are unavailable.

NoteCards runs its own process scheduler; NoteCards is always running as far as the UNIX scheduler is concerned. For this reason, other heavy computational jobs on the same Sun Workstation will not get as good performance as they would competing with conventional UNIX interactive environments.

Similarly, NoteCards may not have adequate interactive performance if it is competing with other computationally-bound processes on the same Sun Workstation.

For these reasons, we recommend that NoteCards be used on machines that are set up primarily for a single user.

Release Contents

The release distribution contains the following documentation and software.

Documentation

The NoteCards documentation kit contains the following:

- *NoteCards User's Guide*
- *NoteCards Release Notes*
- *A User's Guide to TEdit*
- *A User's Guide to Sketch*

- Sun Type 3 and Type 4 keyboard templates

Software

The software release comes on a ¼-inch tape cartridge. The software release is specific to the Sun architecture (Sun 3 or 4) for which you purchased NoteCards, but contains multiple SunOS versions. This tar tape contains the following directories:

<code>./install.sunos3.mc68020</code>	Contains <code>lde</code> , <code>ldeether</code> , <code>ldeether.c</code> , <code>ldesingle</code> , <code>ldesingle.o</code> , <code>ldex</code> , <code>ldex.o</code> , <code>makefile</code> , and <code>usersubrs.c</code> for SunOS 3.x.
<code>./install.sunos4.mc68020</code>	Contains <code>makefile</code> , <code>lde</code> , <code>lde.o</code> , <code>ldeether</code> , <code>ldeether.c</code> , <code>ldemulti</code> , <code>ldemulti.o</code> , <code>ldesingle</code> , <code>ldesingle.o</code> , <code>ldex</code> , <code>ldex.o</code> , and <code>usersubrs.c</code> for SunOS 4.0.
<code>./install.sunos4.sparc</code>	Contains <code>makefile</code> , <code>lde</code> , <code>lde.o</code> , <code>ldeether</code> , <code>ldeether.c</code> , <code>ldemulti</code> , <code>ldemulti.o</code> , <code>ldesingle</code> , <code>ldesingle.o</code> , <code>ldex</code> , <code>ldex.o</code> , and <code>usersubrs.c</code> for SunOS 4.0.
<code>./install.sunos4.lsparc</code>	Contains <code>makefile</code> , <code>lde</code> , <code>lde.o</code> , <code>ldeether</code> , <code>ldeether.c</code> , <code>ldemulti</code> , <code>ldemulti.o</code> , <code>ldesingle</code> , <code>ldesingle.o</code> , <code>ldex</code> , <code>ldex.o</code> , and <code>usersubrs.c</code> for SunOS 4.1.
<code>./sysouts</code>	Contains the <code>sysout</code> , <code>notecards.sysout</code>
<code>./fonts</code>	Contains the font directories.
<code>./checksumdir</code>	Contains <code>ldchecksum</code> , <code>checksum</code> , and <code>X.sum</code> , <code>checksum</code> files and <code>README</code> file.
<code>./ncfiles</code>	Contains <code>DEMO.NOTEFILE</code> and <code>NC-DOCUMENTATION.NOTEFILE</code> .
<code>./patches</code>	Contains various post-release patches to the Notecards system.

Table 2-1 shows the organization of the font directories, as well as the descriptions and contents of the directories.

Table 2-1. Font Directories

Directory Name	Description	Font Families	Font Types
./fonts/display/presentation ./fonts/interpress/presentation	All presentation fonts for display and user interface applications	Helvetica Gacha Times Roman	Sans serif Monospace screen font in 8, 10, 12 MRR Serif
./fonts/display/publishing ./fonts/interpress/publishing	All publishing fonts for character sets, foreign characters, and technical alphabets	Classic Modern Terminal	Serif; in all character sets, sizes, faces Sans serif; in all character sets, faces, but with selected sizes Monospaced, in all character sets, faces, but with selected sizes
./fonts/display/printwheel ./fonts/interpress/printwheel	All printwheel fonts for word processing applications	BoldPS LetterGothic Titan	Proportional serif Monospaced sans serif Monospaced serif
./fonts/display/JIS1 ./fonts/interpress/JIS1	Japanese Kanji fonts, character set 1	Classic	Point sizes 8 through 24
./fonts/display/JIS2 ./fonts/interpress/JIS2	Japanese Kanji fonts, character set 2	Classic	Point sizes 8 through 24
./fonts/display/chinese ./fonts/interpress/chinese	Chinese character fonts	Classic Modern	Point sizes 12 and 24 12 point
./fonts/display/miscellaneous ./fonts/interpress/miscellaneous 30	Miscellaneous fonts for nonstandard and rare applications	ClassicThin Hippo Logo Math OldEnglish Symbol Tonto	Brackets and parentheses in point sizes 16, 20, 26, and Greek Xerox logo Math symbols Point sizes 10 and 18 Math symbols Thick monospaced, 14 point MRR
./fonts/postscript ./fonts/press	All the postscript fonts All metric information for Press printers.		

If you do not use some of these files, you may choose to delete them after installation. Alternately, you might selectively tar off only those files/directories of interest (see your UNIX documentation). For example, most sites might not use Chinese fonts, or

InterPress/Press printers, so the directories `./display/chinese`, `./interpress`, and `./press` would be candidates for deletion.

[This page intentionally left blank]

3. SOFTWARE INSTALLATION

This chapter contains the NoteCards Sun Workstation installation instructions to be used by the system administrator.

To install NoteCards on a Sun Workstation, you need the following:

- Release tape
- NoteCards documentation kit for the Sun Workstation

Ensuring Adequate Swap Space

Prior to installing the software, you should ensure that your system has adequate swap space. NoteCards requires 45 megabytes of swap space on top of the normal swap space requirements. Check swap space using the `pstat` command:

```
prompt% /etc/pstat -s
37176k used (3176k text), 12920k free, 1344k wasted, 0k missing
max process allocable = 10224k
avail: 5*2048k 1*512k 4*256k 3*128k 6*64k 7*32k 7*16k 40*1k
4800k allocated + 2520k reserved = 7320k used, 64672k available
```

If you need more swap space, consult the *Sun Software Technical Bulletin, March 1988*, pages 335-36, for information on increasing the amount of available swap space.

Once you have checked your swap space, you can do the following:

- Load the contents of the release tape
- Configure NoteCards for your system

Installing Software

The installation procedure shows the steps required, to install the NoteCards software, with examples where appropriate. Only those users who are system administrators and have **root** privileges on the host/server where the software is to reside can use this procedure.

Before starting software installation, remember that:

- The software requires 16 megabytes of disk space
- The entire installation procedure takes about 25 minutes to complete on a standalone system with a 1/4-inch tape

1. Log in under your username.

```
login yourname
```

```
prompt%
```

2. Check for available space with the `df (1)` command:

```
prompt% df
```

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/xy0a	7437	5470	1223	82%	/
/dev/xy0h	148455	4900	128709	4%	/usr/misc

3. Determine if you need to run `su` to make a directory for the distribution. If so, type in **su**:

```
prompt% su
```
4. Make a directory for the distribution. If you have enough space on the file system containing `/usr/local`, then create an `lde` directory:

```
prompt# mkdir /usr/local/lde
```

If you don't have enough space on `/usr/local`, go to Step 6.
5. Make yourself the owner of this directory:

```
prompt# /etc/chown yourname /usr/local/lde
```

where *yourname* is your username. Go to Step 7.
6. If you don't have space on the file system which contains `/usr/local`, but do have space somewhere else (for example, on `/usr1`) then make the directory there and link `/usr/local/lde` to it:

```
prompt# mkdir /usr1/lde
prompt# /etc/chown yourname /usr/usr1/lde
prompt# ln -s /usr1/lde /usr/local/lde
```
7. If you ran **su**, type in `exit`:

```
prompt# exit
```
8. Insert the tape cartridge in the cartridge drive.
9. Change the working directory to `/usr/local/lde`:

```
prompt% cd /usr/local/lde
```
10. Load the NoteCards software from tape. Indicate the appropriate device abbreviation for your tape drive by replacing `xx` in the example below with one of the following:
ar for the Archive drive
st for a SCSI tape drive
This example shows the command entry sequence:

```
prompt# tar xvfb /dev/rxx0 126
```

As the software is copied from tape (a process that takes a long time), the system prints a series of lines in the following form:

```
x ./install.sunos4.sparc/lde, 270336 bytes, 532 tape blocks
```

The `x` at the beginning of the line indicates that the file is being extracted from the tape. See Appendix E for information on validating the contents of the tape.

Copy Protection

Before starting NoteCards, you must first obtain a host access key from Venue. This key enables the use of NoteCards on one workstation. The software cannot run without the key. You must have one key for each host on which you wish to run. Note that your current host access key *will* work if you move from one version of SunOS to another.

To obtain a key:

1. Get the host ID of the machine on which you intend to run NoteCards by typing on that machine

```
prompt% hostid
310002f6
```
2. Call Venue at 800-228-5325 between 9:00 a.m. and 4:30 p.m. PST. Outside the United States, call your local distributor.
3. Ask for a host access key, giving the Venue representative your host ID. Venue provides you with a host access key, which you need during software configuration, below.

CAUTION

Depending on your license agreement with Venue, your host access key may have an expiration date. After that date, your key is no longer valid.

Configuring the Software

The software comes in two forms:

- An executable binary image for users who have not modified the Sun kernel too extensively
- An object file that can be relinked for your particular system

If you want to use the executable that Venue supplies, skip to the Enabling PUP/XNS Ethernet section in this chapter.

Relinking

If the prelinked software doesn't work, link the object code with the Sun libraries. To do this, you need the `suntool`, `sunwindow`, and `pixrect` libraries, and `make`, `cc`, etc., available on your search path. To configure the system, connect to the directory `usr/local/lde/install.sunos $x$ .machinex` (where x is the version of SunOS that you are running, e.g., SunOS 4.0 in the following, and *machinex* is the CPU chip, e.g., *sparc* in the following), and type `make`.

```
prompt% cd /usr/local/lde/install.sunos4.sparc
prompt% rm lde ldeether; make
```

This procedure replaces the two executable programs, `lde` and `ldeether`. The program `ldeether` enables access to Xerox network protocols from NoteCards.

Enabling PUP/XNS Ethernet

If you intend to use the PUP or XNS Ethernet directly from NoteCards, you need to change file ownership and permissions of `ldeether`. Note that you do this on the server where `ldeether` is actually residing (root permission must be on the server). Log in to the machine where `ldeether` resides. To find out where `ldeether` resides, type

```
prompt% df filename
```

where *filename* is the pathname of **ldeether**. The system responds with the name of a file system (e.g., `/dev/sd0g`) for a local file, or with a machine name and directory (e.g., `python:/user1`) for an NFS file.

Now you can change the **ldeether** file ownership and permissions.

```
prompt% rlogin server
server% su
server# cd /usr/local/lde/install1.sunos4.sparc
server# /etc/chown root ldeether
server# chmod 4755 ldeether
server# exit
```

If you are using the Ethernet, substitute **ldeether** whenever **lde** appears in the instructions below.

Using/Installing the Host Access Key

Enter the host access key on the command line to run NoteCards:

```
prompt% lde sysoutname -k 'host-key'
```

In this example, replace *host-key* with the actual key (still enclosed in single quotes). Replace *sysoutname* with the name of the sysout you want to use.

You can also avoid having to type the host access key each time. One way is to define an alias in the `.cshrc` file.

```
prompt% alias NoteCards "lde sysout-fullpathname -k 'host-key'"
```

Another method is via a shell script.

```
prompt% echo "lde sysout-fullpathname -k 'host-key'" >  
           /usr/local/lde/install1.sunos4.sparc/NoteCards  
  
prompt% chmod 755 /usr/local/lde/install1.sunos4.sparc/NoteCards
```

Remember to replace *sysout-fullpathname* with the full name of the sysout you want to use. Then, when you wish to run, you can simply type the alias or the script name.

```
prompt% NoteCards
```

Finally, you should arrange for **lde** and **ldeether** to be conveniently located on users' search paths. For example:

```
prompt% ln -s /usr/local/lde/install1.sunos4.sparc/lde  
           /usr/bin/lde  
  
prompt% ln -s /usr/local/lde/install1.sunos4.sparc/ldeether  
           /usr/bin/ldeether  
  
prompt% rehash
```

Rehash recomputes the internal hash table of the directory contents listed in the path variable to allow for new commands.

For more ideas on how to set up your `.cshrc` and `.login` files, see the Starting NoteCards and Exiting NoteCards and Saving State sections in Chapter 4.

[This page intentionally left blank]

4. SYSTEM USE ISSUES

This chapter provides basic system information to get you started in NoteCards on your Sun Workstation after your system administrator has installed NoteCards.

Site Initialization File

When NoteCards starts, it reads in the NoteCards site initialization file, `init.NoteCards`. This file sets the pointer to fonts, site parameters and the like. You need to edit this file if you move the system files from their standard locations. Initialization files are discussed in detail in Appendix D. Unless there are compelling reasons for moving files from their preset directories, we strongly counsel against moving them.

Starting NoteCards

Running NoteCards requires both an executable (`lde`) and a virtual memory image (a `sysout`). Generally, NoteCards requires a host access key matched to the host ID of the processor on which it is running. In addition, if NoteCards is to perform direct network communication (for Xerox PUP or XNS protocols), it needs a separate program, `ldeether`, to initialize the Ethernet. To run NoteCards on the Sun Workstation, follow these steps.

1. Exit `sunttools` or any other windowing system. NoteCards provides its own window system and must not run simultaneously with others.
2. Kill all your user processes (these have console as the control `tty`). Check to make sure you have killed any "selection_ `svc`" process.

Note: If you do not perform this step, certain error messages from UNIX (e.g., "file system full") cause those processes to print to the console, resulting in scrolling of the display .

3. Find out which directory the software is stored on and add that directory to your path if necessary:

```
prompt# set path=($path/usr/local/lde/install.sunosx.machinex)
```

Replace `install.sunosx.machinex` with one of the following:

```
install.sunos3.mc68020
install.sunos4.mc68020
install.sunos4.sparc
```

depending on which machine and SunOS release you are using. You can also add this to your `.login` or `.cshrc` file.

4. Invoke NoteCards by typing the name of the program, e.g.,

```
prompt% lde NoteCards.sysout -k 'this host-key'
```

If you are using either Xerox XNS or Xerox PUP Ethernet protocols, type instead

```
prompt% ldeether NoteCards.sysout -k 'this host-key'
```

This, in turn, runs `lde` and allows it to use the Ethernet directly. As explained in Chapter 3, this command can be replaced by an alias or a shell script.

Where NoteCards Looks for Your Sysout

The system searches for the sysout to be used in the following order.

- *command line*

The name of the sysout file can be given on the command line when starting NoteCards; e.g.,

```
prompt% lde NoteCards.sysout -k 'this host-key'
```

- `LDESOURCESYSOUT`

If no sysout file name is given on the command line, the value of the environment variable `LDESOURCESYSOUT` is used as the name of the sysout file.

- `~/lisp.virtualmem`

Finally, NoteCards looks for a file `lisp.virtualmem` on the home directory of the current user.

Once NoteCards has started, it allocates virtual memory space to hold the working environment, maps the screen frame buffer into its address space, and reads the sysout. As pages from the sysout are read into the display area, they appear on the screen. If the sysout was written while running on a machine with a different size display, the image will appear garbled. This process takes several seconds. After NoteCards starts running, it readjusts the display to the current size.

Exiting NoteCards and Saving State

This section explains how to save a NoteCards system, leave NoteCards, and restart.

On the Sun, `lde` is an ordinary UNIX program that allocates a 45-megabyte data area, reads into that area several megabytes of data (the NoteCards sysout), and modifies it there. Under UNIX, the program's data requirements (which include the sysout) are handled by UNIX; all NoteCards does is modify in virtual memory a copy of your original sysout file. UNIX, transparently to NoteCards, handles all real memory swapping. This has several consequences related to starting, saving, and restarting sysouts.

When you exit NoteCards and save the memory image, you are writing a copy of the virtual memory file, which UNIX has been managing, out to disk. Unlike more usual programs, this file preserves the complete state of the system as it exists when you exit. This means that you can exit a NoteCards session in the middle of editing a file, without saving the file, and when you restart the system the file will be as you left it. However, we always advise you to save your files before you exit.

The virtual memory file that the command **Save image & Exit** writes is normally `~/lisp.virtualmem` (i.e., the file `lisp.virtualmem` on your home directory). However, you can override this default by setting the environment variable `LDEDESTSYSOUT`. For example, you might want to keep virtual memory images on your directory `/users/smith`. You would accomplish this by using the C-Shell command `setenv`. For example:

```
prompt% setenv LDEDESTSYSOUT "/users/smith/lisp.virtualmem"
```

You can also rename the output file as well as its destination. For example, if you wanted the output file to be called `NoteCards.vm`, you would issue the command,

```
prompt% setenv LDEDESTSYSOUT "/users/smith/nc/NoteCards.vm"
```

Place this command in your `.cshrc` or `.login` file to have it executed every time you log in.

In another situation, for instance during a demonstration, you might not want the memory image saved. In this case, you can reset `LDEDESTSYSOUT` to `/dev/null`. For example:

```
prompt% setenv LDEDESTSYSOUT "/dev/null"
```

Cursor tracking interferes with writing out the screen bitmap as part of the `NoteCards` memory image. For this reason, `NoteCards` takes the cursor down before saving a virtual memory image. When this happens, the message

```
Saving VMem, taking mouse down
```

appears in the prompt window, and cursor tracking is disabled.

Because the virtual memory file need not already exist to run `NoteCards`, you can receive the following file errors :

```
File-System-Resources-Exceeded  
Protection-Violation  
File-Wont-Open
```

When working in `NoteCards` on a Sun workstation, you should periodically load a fresh `sysout`. Older `NoteCards` `sysouts` don't run as well as fresh `sysouts` due to a number of factors such as fragmentation of memory, increased working set, and more objects taking up various spaces.

Executing the commands **Save image & Exit** or **Save VM** saves the virtual memory image to a file, which preserves your working state. However, it is important to realize that each time you do this, the file grows in size. Over the course of use, a virtual memory file can easily grow from its original 7-megabyte size to 9 or 10 megabytes.

Keyboard Interpretation

This section describes how NoteCards interprets the Sun Type 3, and Type 4 keyboards. These key assignments were chosen to maximize compatibility with both the Xerox workstation keyboard and the normal Sun keyboards. You can attach a Sun Type 3 or Type 4 keyboard template, which also shows the NoteCards keyboard assignments, to your Sun Type 3 or Type 4 keyboard. Both templates are included with your NoteCards documentation set.

Sun Type 3 Keyboard

Figures 4-1 through 4-3 show NoteCards's key assignments for the Sun Type 3 keypads.

Stop	Again
Help	Undo
Same	Move
Open	Copy
Find	Delete

*Figure 4-1. Sun Type 3
Left Key Pad*

Num Lock	Scroll Lock	Break
7 Home	↑	9 PgUP
4 ←	5	→
1 End	2 ↓	3 PgDN
Ins	DOIT	Caps Lock

*Figure 4-2. Sun Type 3
Right Key Pad*

Center	Bold	Italic		Case		Strikeout		Underline		Super Sub	Larger Smaller	Margin	Back Word
Esc	! 1	@ 2	# 3	\$ 4	% 5	↑ 6	& 7	* 8	(9	) 0	← -	+ =	 \ ~ ,
Tab	Q q	W w	E e	R r	T t	Y y	U u	I i	O o	P p	{ [	}]	Backspace
Ctrl	A a	S s	D d	F f	G g	H h	J j	K k	L l	:	; "	'	Return
Shift		Z z	X x	C c	V v	B b	N n	M m	< ,	> .	? /	Shift	LF
Caps	Meta	Space										Expand	Next

Figure 4-3. Sun Type 3 Center Key Pad Interpretation

Sun Type 4 Keyboard

Figures 4-3 through 4-5 show the keyboard and the left and right key pads for the Sun Type 4 keyboard.

Stop	Again
Props	Undo
Same	Copy
Open	Move
Find	Delete
Help	

*Figure 4-3. Sun Type 4
Left Key Pad*

Break	PrSc	scroll lock	num lock
=	/	*	-
7 Home	8 ↑	9 PgUP	+
4 ←	5	6 →	
1 End	2 ↓	3 PgDN	DOIT
Ins		Del	

*Figure 4-4. Sun Type 4
Right Key Pad*

F1 Center	F2 Bold	F3 Italic	F4 Case	F5 Strike	F6 Under	F7 Super	F8 Large	F9 Margin	F10	F11	F12	 \ _	Delete Word	
Esc	! 1	@ 2	# 3	\$ 4	% 5	↑ 6	& 7	* 8	(9	) 0	← -	+ =	Back Space	
Tab	Q q	W w	E e	R r	T t	Y y	U u	I i	O o	P p	{ [	}]		Return
Ctrl	A a	S s	D d	F f	G g	H h	J j	K k	L l	: ;	" ,	~ '		
Shift		Z z	X x	C c	V v	B b	N n	M m	< ,	> .	? /	Shift	LF	
Caps	Meta	Left Spc	Space									Right Spc	Expand	Next

Figure 4-5. Sun Type 4 Center Key Pad Interpretation

Note: In SunOS 4.0, the NEXT (ALT/GRAPH) key on the Type 4 keyboard is inaccessible. Later versions of SunOS are believed to fix this.

Console Messages

Under SunOS, various system processes and operations attempt to log information on the console. Since NoteCards takes over the screen, console messages are redirected; a background process in NoteCards causes them to appear in the prompt window.

When NoteCards is run remotely (i.e., not from the console), most console, or operating system, messages are printed in the prompt window. Some messages may also appear in the middle of the NoteCards display screen or on the remote tty. This occurs because UNIX is often confused about where to send messages. NoteCards is normally run remotely only for debugging purposes.

CAUTION

Critical UNIX system processes can hang if the buffer holding console messages fills. There are two points to watch for:

NoteCards uses a temporary file, `/tmp/XXXX-lisp.log`, where XXXX is the user's login name, to buffer console messages before printing them. Do not delete this log file while NoteCards is running. If the log file is deleted, console messages can no longer be printed in the NoteCards prompt window.

The process `\10MBWATCHER`, used to watch for Ethernet packets, reads console messages. Thus, you should never kill the `\10MBWATCHER` process, even if you don't use the Ethernet capabilities of NoteCards. If you do kill the `\10MBWATCHER` process, console messages cannot be printed and the operating system can hang.

File Compatibility

Sysout Compatibility between Sun and Xerox Workstations

Sysout files created on Xerox workstations can be moved to Sun Workstations. However, sysout files created on Sun Workstations cannot be moved to Xerox workstations.

File Compatibility between Sun and Xerox Workstations

Some care must be taken in moving files to and from Xerox workstations and services, since the default end-of-line convention in UNIX is to terminate lines with the line feed (LF) character, while traditionally Xerox systems have terminated lines with the carriage return (CR) character. In particular, if you use some other file transfer mechanism, such as FTP or Kermit, be careful to transfer `.TEDIT` and `.NOTEFILE` files in binary mode.

In NoteCards on the Sun Workstation, the default end-of-line convention for all text files is line feed (LF). The default end-of-line convention for all binary files is carriage return (CR); this is because CR is used internally in the system.

When working with the Xerox protocols XNS and PUP as well as Sun NFS, it is important that you use the suffix `.TEDIT` on all TEdit files, the suffix `.SKETCH` on all Sketch files, and the suffix `.NOTEFILE` on all notefiles for NoteCards to treat the files properly when moving them from one device to another.

TEdit files not transferred in binary mode frequently get "Font not found." error messages when you try to open them. The carriage returns, which are represented by the number 13, are reinterpreted as font information and the system breaks trying to look for a font of size 13. These files are irrecoverably lost.

Using SunOS Files from NoteCards

You can access any mounted SunOS file system directly from NoteCards. The mounted file system is available as an I/O device of the NoteCards environment. This file system appears as the local disk of NoteCards, even though it may be a remotely mounted file system of networked Sun file servers.

Many of the file devices to which the NoteCards environment can talk, including PUP, XNS file servers, the {CORE} device, and others, have facilities that are not directly supported by SunOS. For example, many file systems have file version numbers and case insensitive file search conventions.

NoteCards on the Sun Workstation has two distinct "host" names that can be used to access the SunOS file system. These host names are provided for compatibility with existing applications and tools. They also simultaneously allow natural interaction with the SunOS file system. The names are:

- {DSK} On the Sun Workstation, the {DSK} device allows you to access the file system using similar conventions to those used for Xerox services. In particular, {DSK} files have version numbers; {DSK} file name recognition also ignores the case of letters.
- {UNIX} The {UNIX} device lets you use the mounted file systems with the normal naming conventions of the SunOS file system. {UNIX} files do not have version numbers, and the file name recognition treats lowercase letters as distinct from their uppercase equivalents.

File streams can be opened or closed on both devices. The reason for having both devices is to more easily support the running of applications that were originally developed using Xerox services, while still allowing new applications to interact more naturally with UNIX.

File Naming Conventions

In NoteCards, a file name (pathname) consists of a collection of fields: the *host*, *directory*, *name*, *extension* and *version*. These fields are optional. The standard NoteCards syntax for these fields is:

```
{host}<directory>name.extension;version
```

The *directory* field can be a directory path consisting of a sequence of directory and subdirectory components. Slashes (/) and right angle brackets (>) can be used to delimit a directory name; there is no distinction made between them. However, you should use one delimiter or the other consistently. What the directory returns is unpredictable if you mix delimiters. Note that square brackets ([]) are not acceptable as directory delimiters.

Duplicated directory delimiters are treated as a single delimiter. Thus, the following two file names specify the same file:

```
{DSK}<LISP>USERS>FOO.;1  
{DSK}</LISP/USERS/>FOO.;1
```

Common {DSK} and {UNIX} Naming Conventions

- To include a special character (e.g., > or ;) in a file name, precede it with a single quote ('). To include a single quote in a file name, precede it with another single quote. You can quote any of these characters: <, >, ;, ~, and a period (.). The following examples show how the single quote notation on {DSK} and {UNIX} is used.

{DSK} Name From Lisp

```
foo'>bar.baz;1
foo';bar.baz;1
foo''bar.baz;1
```

File Name From SunOS

```
foo>bar.baz
foo;bar.baz
foo'bar.baz
```

- {DSK} and {UNIX} do not allow you to use either the slash (/) or the NUL character in file names. Thus, you cannot name files containing these characters.
- Both {DSK} and {UNIX} can handle the following characters, which were defined as special characters in Medley Release 2.0: backslash (\) and tilde (~).
- {DSK} and {UNIX} can distinguish between a file name with a period at the end (e.g., foo.) and a simple file name (e.g., foo). The final period is preceded with a single quote, as shown in the following example:

{DSK} Name From Lisp

```
foo.;1
foo'..;1
```

File Name From SunOS

```
foo
foo
```

- On {DSK} and {UNIX}, the C-Shell and SunOS directory notations (~, ., and ..) are supported in the Notecards directory specification. The tilde character (~) is allowed at the very beginning of the directory specification of a pathname. A combination of relative path specifiers (~, ., ..) is supported. The tilde character corresponds to the user's home directory at login. The period (.) corresponds to the current working directory. Two periods (..) indicates the parent of the current working directory.
- File names are returned by the system (e.g., INFILEP) in more canonical form. The function which returns the full file name returns it in the canonical form, as in {DSK}<usr>etc rather than {DSK}/usr/etc/. This change will make some tools which depend on the conventional file name representation described in the *Interlisp-D Reference Manual* work correctly on the Medley file system (e.g., COPYFILES).

{DSK} Naming Conventions

File access to the {DSK} device goes through the following file name transformation when actually accessing the SunOS file system:

- Mixed case letters are read as such.
- File name searches are done case-sensitive first; if a match is not found, the system then does a case-insensitive search.
- A left angle bracket (<) is translated to a slash (/), the delimiter for the root directory.
- {DSK} supports relative pathnames. You can specify relative pathnames by omitting a slash (/) or left angle bracket (<) as the first character in the directory field. For example:

`{DSK}foo.fee` and `{DSK}~/foo.fee` are relative to your UNIX home directory (`~/foo.fee`).

`{DSK}./foo.fee` is relative to your current working directory (`./foo.fee`).

`{DSK}../foo.fee` is relative to the parent directory of your current UNIX working directory (`../foo.fee`).

The Medley 2.0 `{DSK}` device supports the notation in which the three meta characters (`'`, `...`, and `~`) are used together, as shown in the following example:

```
{DSK}~/../tom/foo.c
```

In this example, the `{DSK}` device interprets `tom` as one of the subdirectories of the parent directory of the user's home directory.

`{DSK}` also supports the tilde-name (`~name`) convention. `{DSK}` interprets `{DSK}~tom/foo.c` as a file named `foo` on `tom`'s home directory. In this notation, the user name is case-sensitive (e.g., `~tom` and `~Tom` are treated as different users).

Version Numbering

The UNIX file system does not support version numbers in file names; `{DSK}` emulates versions with a naming convention. This section explains how `{DSK}` version numbers are represented in the SunOS file system.

- When you create a completely new file, it appears in the SunOS file system without a version number.

{DSK} Name From NoteCards	File Name From SunOS
----------------------------------	-----------------------------

<code>bar.baz;1</code>	<code>bar.baz</code>
------------------------	----------------------

- When you create (from NoteCards) a file with a version other than 1, NoteCards adds version numbers to that file name, as a trailing number between tildes, e.g., "`myfile.~12~`" for the twelfth version of `myfile`.

The following shows some examples of equivalent file names in NoteCards and SunOS.

{DSK} Name From NoteCards	File Name From SunOS
----------------------------------	-----------------------------

<code>bar.baz;1</code>	<code>bar.baz.~1~</code>
<code>bar.baz;2</code>	<code>bar.baz.~2~</code>
<code>bar.;23</code>	<code>bar.~23~</code>

- NoteCards always maintains a versionless file which is hard-linked to the highest extant version of the file (i.e., they are two names for the very same file). This file name does not appear in the `{DSK}` directory listing.

From {DSK}

`foo.c;15`
`foo.c;23`

From SunOS

`foo.c` (hard linked with `foo.c.~23~`)
`foo.c.~15~`
`foo.c.~23~`

Similarly, a file created in UNIX with no version number is treated by `{DSK}` as the highest version.

- When you create a new version of a file, the versionless-file link is broken, and the versionless file is hard-linked to the new highest version.

From {DSK}

foo.c;15
foo.c;22
foo.c;24 (new file)

From SunOS

foo.c (hard linked with foo.c.~24~)
foo.c.~15~
foo.c.~22~ (no link with foo.c)
foo.c.~24~ (new file, link from foo.c)

- When you delete the highest version of a file, the versionless file is also deleted. If any older versions of the file remain, a new link is created from the versionless name to the highest version extant. For example, if you have the files

From {DSK}

foo.c;1
foo.c;2

From SunOS

foo.c (linked to foo.c.~2~)
foo.c.~1~
foo.c.~2~

and you delete foo.c;2 from {DSK}, the resulting files are:

From {DSK}

foo.c;1

From SunOS

foo.c (linked to foo.c.~1~)
foo.c.~1~

- When you rename a file it works the same as deleting the file under the old name then creating it under the new name. For example, if you have the following {DSK} files,

From {DSK}

foo.c;1
foo.c;2
fee.c;1
fee.c;2

From SunOS

foo.c (linked to foo.c.~2~)
foo.c.~1~
foo.c.~2~
fee.c (linked to fee.c.~2~)
fee.c.~1~
fee.c.~2~

and you rename "foo.c" to "fee.c", your renamed {DSK} files and the linked SunOS files would appear as

From {DSK}

foo.c;1
fee.c;1
fee.c;2
fee.c;3

From SunOS

foo.c (linked to foo.c.~1~)
foo.c.~1~
fee.c (linked to fee.c.~3~)
fee.c.~1~
fee.c.~2~
fee.c.~3~ (renamed file)

- When a file has a name suffix that is not a valid version number (e.g., myfile.~12x~), the suffix is regarded as part of the file name.

From {DSK}

myfile.~12x~;1

From SunOS

myfile.~12x~

Pathnames

A pathname on {DSK} is always case insensitive. When you specify a file, the {DSK} device handler first searches for the file with the specified name. If no such file is found, it then searches for a file with the same spelling but different case.

If a pathname on {DSK} has no directory specification, a tilde-slash combination (~/) is used, i.e., the NoteCards directory specification {DSK}foo is the equivalent of UNIX ~/foo.

{UNIX} Naming Conventions

For the {UNIX} device, file name translation takes place only on the directory. An initial left angle bracket (<) is treated as if it were an initial slash (/); both signify a path relative to the SunOS file system root directory; if there is no initial left angle bracket or slash, the directory is relative to the current working directory. Initially this is the working directory where NoteCards was started; you can change it using the CHDIR function, described below. Tilde (~) translates to your home directory.

For example,

```
{UNIX}myfile/abc
```

means the file abc on the ./myfile/ directory.

The {UNIX} device does not recognize version numbers, does not return them, and ignores them for recognition.

No case translation or recognition is done; upper- and lowercase letters are treated as distinct.

Examples:

{UNIX} Name From NoteCards	File Name From SunOS
<foo>fee>bar.baz;1	/foo/fee/bar.baz;1
<foo>fee/bar.;1	/foo/fee/bar.;1
<foo/fee>	/foo/fee/
</foo/fee/>	/foo/fee/
/foo/fee/bar.~1~	/foo/fee/bar.~1~
/foo/fee/	/foo/fee/

In the first two examples the ;1 is treated as part of the file name, not the version number. Note in the last two examples that translation is not done.

Directories

In places where NoteCards expects a directory name, {UNIX} paths must end with a slash (/).

Directory Enumeration

You cannot use the wildcard character, asterisk (*), in subdirectories for either {DSK} or {UNIX} devices. For example, typing

```
{DSK}/users/x*/foo
```

into the FileBrowser prompt window will return

```
No files in group {DSK}<users>x*>foo.*;*
```

Enumeration of files in directories differs between {DSK} and {UNIX} devices. On the {DSK} device, a versionless file which has a link to the highest version file is not enumerated in a directory.

On the {UNIX} device, all files are enumerated in a directory. For instance, if the following SunOS files, linked with `foo.c.~2~` exist

```
foo.c
foo.c.~1~
foo.c.~2~
```

the {DSK} directory enumeration, would look like this:

```
> (DIRECTORY ' {DSK}/users/venue/*)
({DSK}/users/venue/foo.c;1
{DSK}/users/venue/foo.c;2)
```

The {UNIX} directory enumeration, on the other hand, would look like this:

```
> (DIRECTORY ' {UNIX}/users/venue/*)
({UNIX}/users/venue/foo.c
{UNIX}/users/venue/foo.c.~1~
{UNIX}/users/venue/foo.c.~2~)
```

Directory Creation

{DSK} When you write a new file on {DSK}, if the directory named in a pathname does not exist, the {DSK} device handler creates the directory automatically. This feature is provided for compatibility with Xerox services.

If you try to connect to a nonexistent directory (using the Directory Connector tool), NoteCards beeps and prints the following message in the system prompt window

```
Not a valid directory name.
```

{UNIX} The {UNIX} device does not support such directory creation. An attempt to create a file on a nonexistent directory results in an error. NoteCards beeps and prints the following message in the system prompt window.

```
System call error: stat errno = 2
No such file or directory.
```

Directory Deletion

Neither {UNIX} nor {DSK} support automatic directory deletion. To delete a directory you must use the SunOS C-Shell command `rmdir`.

Open File Limit

The number of simultaneously open {DSK} and {UNIX} files must fall within the SunOS limits for a process. For OS 3.4, this number of open files may be configured, with 30 as the maximum permissible number of open files per process. This means that it is not possible to have more than 30 files open for a process, minus whatever files NoteCards has open for its own use, at any one time in the NoteCards system. If you try to open too many files, the system call error number 24,

```
Too many open files
```

appears in the prompt window.

For OS 4.0, the maximum number of files/processes that can be open at one time is 64, unless your kernel is configured otherwise.

Default Pathname

If no path is given, the {DSK} device defaults to the user's home directory, tilde-slash (~/). The {UNIX} device defaults to the current working directory. This current working directory can be changed with the Directory Connector tool. Note that the current working directory is also used to resolve the interpretation of the period (.) and double period (..) specifications at the beginning of a {DSK} pathname. The current working directory is the directory you were in when you started NoteCards, not your connected directory in NoteCards.

File Attributes

This section describes how the various file attributes are treated by NoteCards on the Sun Workstation and what they translate to in SunOS. For more discussion of file attributes, see Chapter 14 for a description of the FileBrowser.

WRITEDATE and CREATIONDATE

[File Attributes]

The date is reset to the current time whenever the contents of a file are modified. Since UNIX does not naturally support more than one date for file modification, the WRITEDATE and CREATIONDATE are treated identically by NoteCards, and by the {DSK} and {UNIX} devices.

TYPE

[File Attribute]

Returns the TYPE property of files; normally either TEXT or BINARY. However, UNIX does not distinguish between TEXT and BINARY files. Normally, programs will infer the type by the file extension. If no file extension is given, the default file type TEXT is used.

SIZE

[File Attribute]

Returns the file size. For compatibility with other NoteCards environments running on Xerox workstations and servers, the SIZE attribute is computed as the length of the file (in bytes) divided by 512 (rounded up).

File System Errors

Several types of errors may occur in the NoteCards file system.

When a remotely mounted file system or the NFS service is down, any attempt to access a file on the file system eventually results in a timeout error. The following error message is printed in the prompt window:

```
File access timed out
```

If the mounted device is mounted with the "hard" option, NoteCards continues to wait until the mounted device responds. During that time, user interrupts are not available. We recommend mounting remote file systems with the "soft" option. You can use the UNIX command `/etc/mount` to check the current mount options.

The following error messages may appear when there are NoteCards file errors:

```
Not owner
Device error:
Protection-violation
File-won't-open
Too-Many-Files-Open
File too large
File-System-Resources-Exceeded
Connection timed out
No-Such-Directory
Bad Host Name
FS-RENAMEFILE-SOURCE-COULDNT-DELETE
```

Another type of error occurs occasionally when the file system prints an incorrect message such as

```
File not found
```

A more accurate console message appears, at the same time, in the prompt window. This message appears in the SunOS message form

```
System call error:  open errno=13  Permission denied
```

See the *UNIX Interface Reference Manual*, Intro (2), for descriptions of all OS system call messages.

[This page intentionally left blank]

5. NOTECARDS BASICS

This chapter provides the basic introduction to NoteCards. This is the place to start, after your system administrator has installed your system, if you need a quick introduction to NoteCards before diving into the body of the reference manual. This chapter explains how to:

- Use the keyboard and the mouse
- Deal with simple error conditions
- Start NoteCards
- Exit NoteCards
- Use NoteCards

A Few Pointers on Mouse Etiquette

Once you have completed the software installation and booted the system so that it is up and running, as explained in Chapters 3 and 4, you are ready to start a NoteCards session. The NoteCards system is running when you can access the mouse cursor. When the system starts, the mouse cursor will appear in the upper left corner of the screen. Take the mouse and move it around on the mouse pad in circles until you see the mouse cursor moving.

On both the Sun and Xerox workstations, the mouse is an optical device. Which is to say that it reads the patterns on the mouse pad. The Xerox mouse does not care how you position the mouse pad; some Sun mice do. If the mouse behaves strangely, turn the mouse pad so that it is positioned as shown in Figure 5-1.

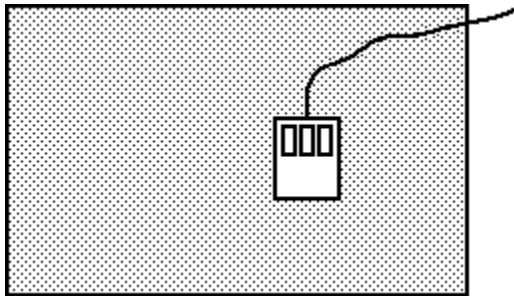


Figure 5-1. Relative positioning of Mouse and Mouse pad for Sun Workstations

Sun machines also have both high- and low-resolution mice. So if the mouse does not work at all and you have verified the connections, you might try a different mouse pad. The high-resolution mouse pads are smaller than the low-resolution mouse pads, but the mice look the same.

Mouse Button Use

The result of pressing a mouse button is fairly consistent throughout the system. The following are general rules for mouse button use. Examples follow in the next section.

Left Mouse Button

Used for selecting and positioning items.

In the window title bar of cards, pops up the NoteCards system menu.

In text windows, the left mouse button selects small structures (characters & lines).

Middle Mouse Button

Frequently brings up a menu associated with a particular window or tool such as a TEdit window or a clock window.

In text windows, the middle mouse button selects larger structures than the left mouse button (words & paragraphs).

Right Mouse Button

Pops up the system background, window, and icon menus.

In text windows, the right mouse button extends previous left- and middle-mouse button selections.

Regions Sensitive to Mouse Activity

There are certain regions of the screen and windows which are sensitive to mouse activity or respond to mouse buttons being pressed when the mouse is over them.

Background

The background is the gray part of the screen which the windows sit on top of. Figure 5-2 shows a snap of the NoteCards MenuBox Icon on the background.

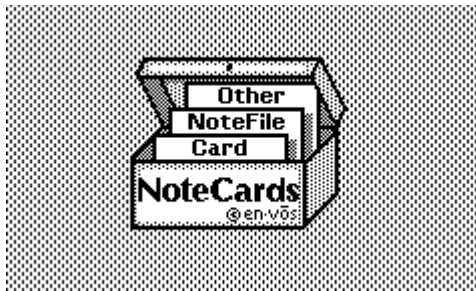


Figure 5-2. NoteCards MenuBox Icon on the Background

The background menu is one of the system menus. To bring up the background menu, move the mouse cursor *out of any windows or icons*, onto the gray background, and press and hold down the right mouse button. When you do this, the menu will pop up. (For this reason, these menus are known as "pop-up" menus.) To release the menu, slide the mouse cursor away from the menu and release the mouse button.

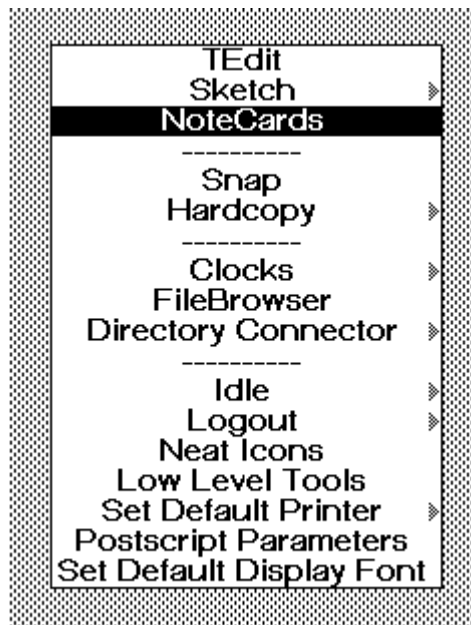


Figure 5-3. **Background** Menu with **NoteCards** Option Selected

To select an option from the background menu:

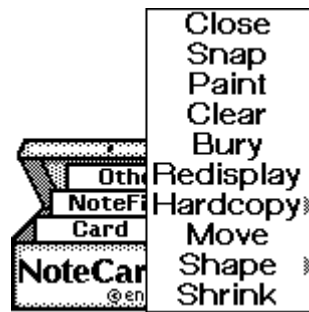
1. Press and hold the right mouse button in the background
2. Still holding down the right mouse button, slide the mouse cursor over the background menu.
3. Move the cursor up and down until the option you want appears in reverse video *and only then* release the right mouse button.

Note: **When you release the right mouse button be very careful not to slide off your selected menu option onto another menu option.** The cursor "hot spot" or the point on the cursor that the menu is sensitive to is the very tip of the cursor arrow.

Try selecting the **NoteCards** option now. When you select it, the MenuBox icon should flash. If you close the MenuBox Icon, as you will do shortly, you can reopen it by selecting the NoteCards option from the background menu.

Windows and the Window Title Bar

The MenuBox Icon is a window and has the usual window menu. Place the mouse cursor on the MenuBox and press and hold the right mouse button down. When you do this, the window menu will pop up as shown in Figure 5-4.

Figure 5-4. **Window** Menu

For some windows, e.g. text windows, the right mouse button has different uses in the body of the window and does not bring up the window menu. In these cases, you gain access to this menu by positioning the mouse cursor on the window title bar before you press the right mouse button.

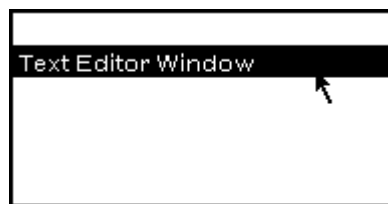
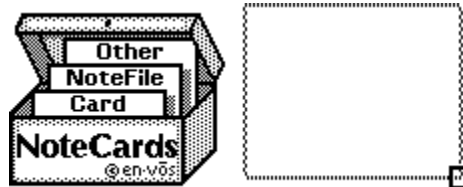


Figure 5-5. Cursor in Window Title Bar of a TEdit Window

Note that it is the tip of the mouse cursor arrow which is the hot spot. The system reads the cursor as being at this particular pixel (point) on the screen.

You select options from the window menu the same way you select options from the background menu. Select **Close** and then reopen the MenuBox Icon from the background menu as discussed above.

Next, select **Move**. An outline of the window will appear, which you can position by moving the mouse. Plant the window by pressing and releasing the left mouse button.

Figure 5-6. **Move** Command Prompting for new MenuBox Icon Position

Move the mouse cursor to the black window on the screen. This is the system prompt window. System status information is printed out here. Select **Shape** from this window's right button menu and the mouse cursor takes on the following form.



Figure 5-7. Mouse Cursor Prompting Sweep-Out of Window Size and Shape

To reshape the window, position the cursor where you want the upper left corner of the window to be, press and hold down the left mouse button, sweep out a new region for

the window, and release the left mouse button when the window has the size and shape you want. More techniques for shaping windows are discussed in Chapter 7.

Select **Shrink** to shrink a window down to an icon or a title bar. Try shrinking the MenuBox Icon. Its shrink icon is shown below.



Figure 5-8. MenuBox Shrink Icon

Icons

You expand a shrink icon by clicking the middle mouse button inside the icon. You can move a shrink icon by pressing and holding down the left mouse button while moving the shrink icon to its new position. Release the left mouse button when the icon is where you want it.

Pressing and holding the right mouse button down in an icon pops up the **Icon** menu.

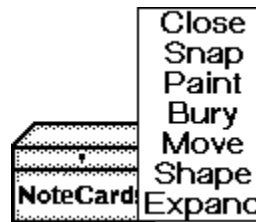


Figure 5-9. Icon Menu

Scroll Bars

Scroll bars automatically appear when you slowly slide the mouse cursor from inside the window to just outside of the left or bottom side of a window when there is more in the window than can be displayed on the screen.

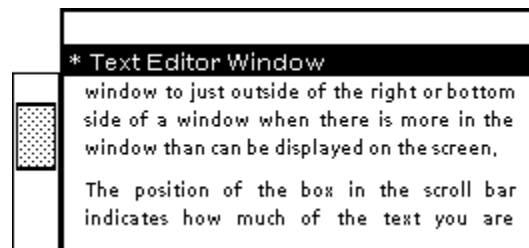


Figure 5-10. TEdit Window with Scroll Bar

The position of the box in the scroll bar indicates how much of the text you are looking at. You scroll the window up by clicking or holding the right mouse button down. You scroll the window down by clicking or holding the left mouse button down. You thumb or jump to a particular point in the document by pressing the middle mouse button.

This gives you a brief overview of how to get around in the system using the mouse. We will cover these and other mouse operations, in more detail, as we walk you through NoteCards.

If ever you start something you don't want to finish, just hit the STOP key. On the Sun keyboard it is the key labelled "L1." This will abort you out of most operations.

Dealing with Simple Error Conditions

If you tried some of the other menu options skipped over above, you may have run across something called a "Break Window." This is most likely to have happened if you chose the **Hardcopy** option, without having first told the system about your printers.

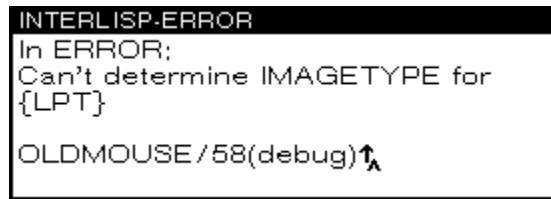


Figure 5-11. Break Window

Break windows are used by people extending the NoteCards system. As a user, you will want to up-arrow out of the break and either try the operation a second time, or check to make sure that your system is properly configured before trying again. For how to configure your system, see Appendix D.

To up-arrow out of a break, click the left mouse button in the window to pass it the type-in process, shown by the flashing caret, and type a ^ followed by a RETURN. The ^ is the character you get when you hold the shift key down and press the "6" key.

Starting NoteCards

Start NoteCards by opening a notefile. A demo notefile is distributed with the system, and this is what we will be working with in the rest of this chapter. If you have followed the instructions in Chapter 3 for installing the system, the demo notefile is located in the directory {dsk}/usr/local/lde/lisp/.

Opening a Notefile

Open the demo notefile by pointing the mouse cursor at the **Notefile** option on the MenuBox Icon and pressing and holding down the left mouse button. When you do this the Notefile Ops menu will pop up. Holding down the left mouse button, slide the mouse so that the **Open** option appears in reverse video, as shown below, and then release the left mouse button.

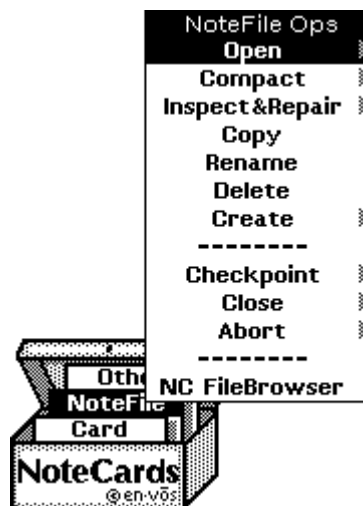


Figure 5-12. Notefile Ops Menu with the **Open** Option Selected

Once you have completed this operation a prompt window opens on top of the MenuBox. Type the path name of the demo notefile. It is the same as shown in the example unless it has been moved to another directory. If you have plenty of file space, you might want to make a copy of the demo notefile in your own directory so that you can modify it as much as you like and still have the original notefile available.

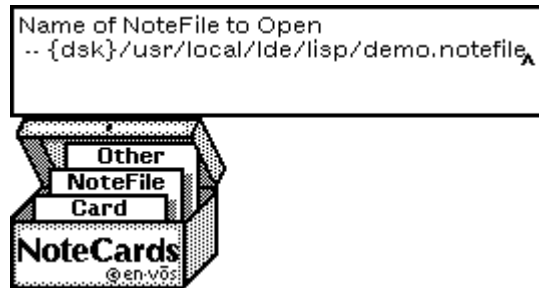


Figure 5-13. MenuBox Icon Prompting for Name of a Notefile to Open

Once you have entered the name of the notefile followed by a carriage return, the system will remember the notefile name so you will not have to type it again.



Figure 5-14. NoteCards prompting for Notefile Banner Position

After NoteCards has located the demo notefile, it asks you to position the notefile Banner on the screen. Move the position box to an appropriate location and click the left mouse button to plant it. The demo notefile Banner then appears.



Figure 5-15. Demo Notefile Banner

The Banner is your interface to the notefile. There is one Banner on the screen for each notefile which you have opened. Once you can access the Banner, the notefile is open.

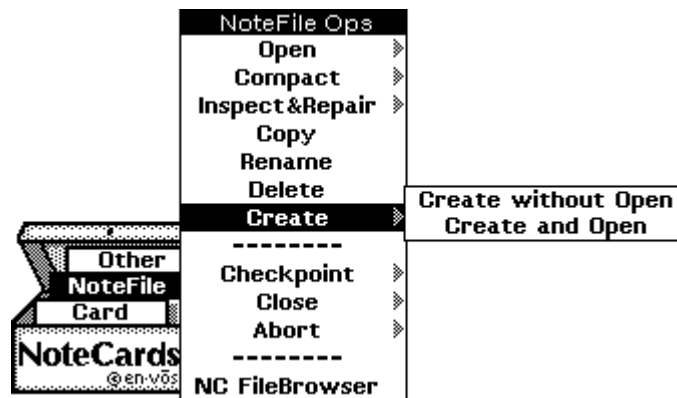
After you have positioned the notefile Banner a second ghost box (like the one shown in Figure 5-14) appears to prompt you to position the demo notefile "Table of Contents" FileBox. This is the top level FileBox in all notefiles. Once this FileBox is open you are now ready to explore.

demo;1 {DSK}/usr/local/ide2/lisp/demo	
Table of Contents	
FILE BOXES	
☒	NoteCards Model
☒	Examples of Notecard Types
☒	NoteCards Applications
☒	NoteCards Functionality
☒	Introduction to NoteCards
☒	Interactive Demonstration Script
NOTE CARDS	
☐	How to use this demo

Figure 5-16. Demo Notefile "Table of Contents" FileBox

Creating a New Notefile

To create a new notefile, go to the MenuBox Icon, press and hold down the left mouse button on the **Notefile** option and select **Create** from the Notefile Ops menu. The default is **Create without Open**. However, if you want to perform the create and open operations together you can slide off and select the **Create and Open** option on the **Create** submenu.

Figure 5-17. **Create** Option and its Submenu on the **Notefile Ops** Menu

The system next prompts you for the name of the new notefile.

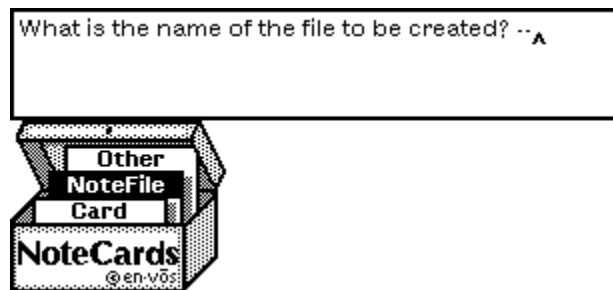


Figure 5-18. MenuBox Icon Prompting for Name of New Notefile to Create

If you only provide the notefile name, the new notefile will be stored in your connected directory. The notefile name traditionally has a **".notefile"** suffix.

If you are unsure as to your connected directory, you can bring up the directory connector tool which will constantly display and allow you to change your connected directory.



Figure 5-19. Background Menu **Directory Connector** Option

To start the directory connector, place the mouse cursor in the background and press and hold the right mouse button. Still holding down the mouse button, position the cursor over the Directory Connector option and release the mouse button. A directory connector window will appear in the lower left corner of your display, showing you your currently connected directory.

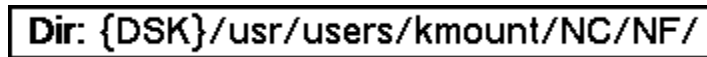


Figure 5-20. Directory Connector Window

Closing Notefiles and Exiting NoteCards

Before we start exploring the demo notefile, let me first explain how to close a notefile and exit the system so that if you need to stop to do something else on your machine you know how to leave and reenter NoteCards.

Closing Notefiles

You access the close-notefile menu from the notefile's Banner title bar. Position the tip of the mouse cursor in the title bar and press and hold the left mouse button down. A short version of the **NoteFile Ops** menu will pop up.

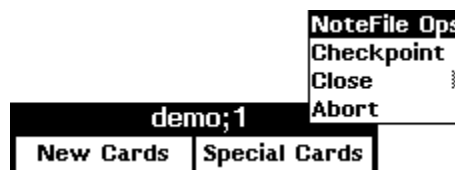


Figure 5-21. **NoteFile Ops** Menu fromBanner of Open Notefile

There are two ways to close a notefile.

Close

Saves all the changes you have made to the notefile and then closes the notefile.

Abort

Discards all the changes you have made to the notefile and then closes the notefile. This option is your best choice when you are working on the demo notefile as it will

allow you to make as many changes as you want and then undo them when you close the notefile.

When a notefile is closed its Banner changes slightly, The **New Cards** and **Special Cards** menu options gray over, and it has a different **NoteFile Ops** menu.

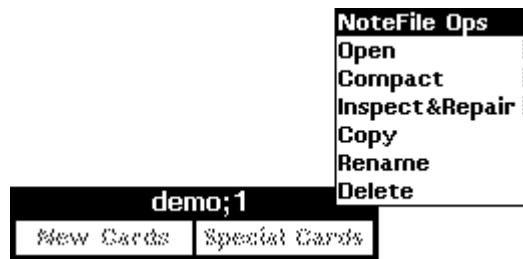


Figure 5-22. **NoteFile Ops** Menu from Banner of Closed Notefile

Exiting NoteCards

There are three ways to get back to UNIX from NoteCards. Each leaves the NoteCards system in a slightly different state.

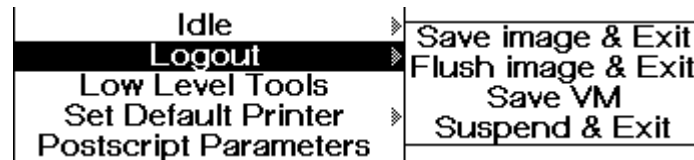


Figure 5-23. **Background Menu Logout** Submenu Options

Logout / Save image & Exit

Save the NoteCards system state and return you to UNIX. Saving the system state involves writing the current system state to a file on disk. The name of this file is either `lisp.virtualmem` on your home directory or the path and file name specified by the UNIX environment variable `LDEDESTSYSOUT`.

Writing out the virtual memory file takes several minutes but exiting in this fashion means that the system will restart in exactly the same state you left it in when you logged out.

Flush image & Exit

Returns your system to UNIX without saving the working image.

Suspend & Exit

Pops you out to UNIX without damaging the NoteCards image. Use this option when you want to do something quickly in UNIX and then return to NoteCards. To return to NoteCards type the command **fg** (foreground) at the UNIX prompt.

Using NoteCards

If the "Table of Contents" FileBox is not open on the screen, go to the demo notefile Banner (Figure 5-15), and click the left mouse button on the Special Cards option. A

ghost box representing the "Table of Contents" FileBox will appear which you should position on the screen and then plant by clicking the left mouse button.

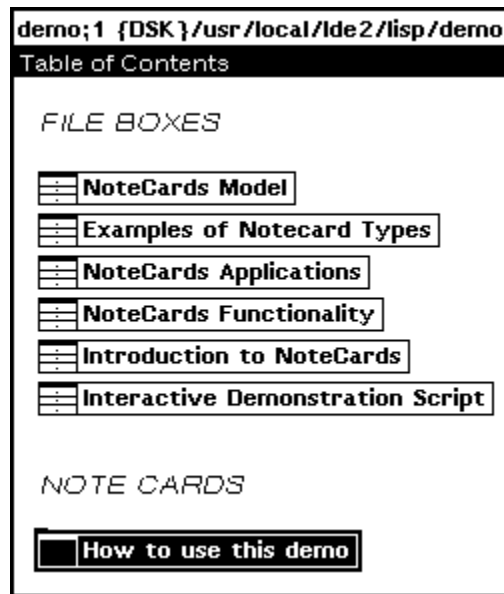


Figure 5-24. Demo Notefile "Table of Contents" FileBox with "How to use this demo" Link Icon Selected

Opening Cards / Traversing Links

To follow a link from one card to another, click the left mouse button on a link icon. In some instances it can take two clicks to traverse a link. In this case, the first click designates the card as the active card and the second click traverses the link. You have successfully accessed a link when the link icon appears in reverse video, as shown in Figure 5-24.

Traverse the "How to use this demo" link in the "Table of Contents" FileBox to open the card shown below. And use this card as the start point for your explorations of NoteCards.

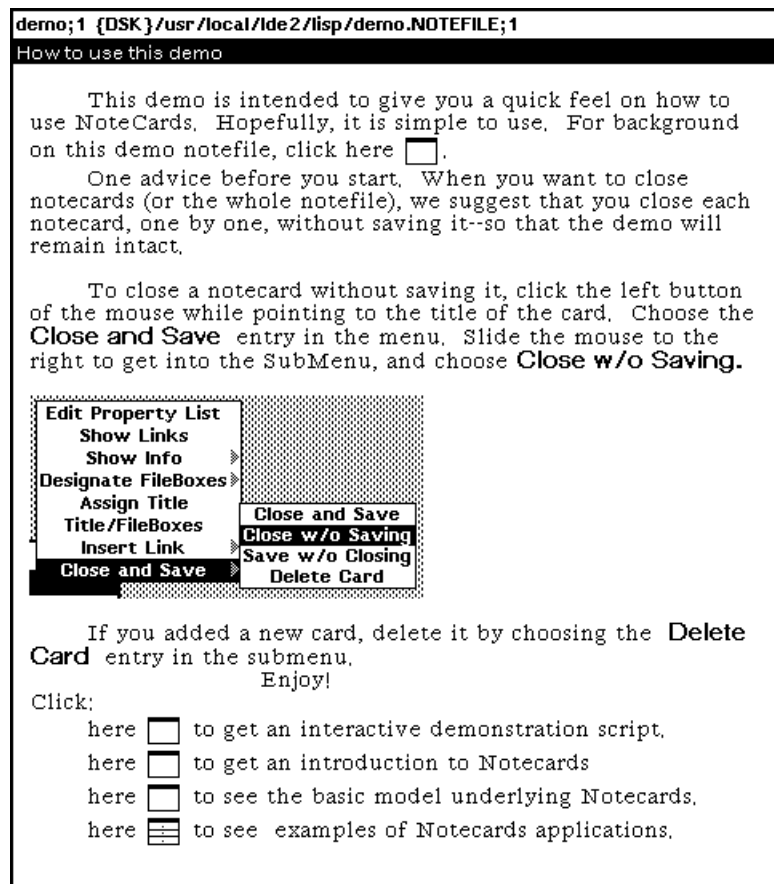


Figure 5-25. "How to use this demo" Card

Shrinking Cards

As you work with Notecards, even on a high resolution screen, you will find that the screen fills up with cards and you will need to close some or move them out of the way.

You can shrink cards down to icons by pressing the right mouse button down in the title bar of a card and selecting the **Shrink** option. The card will be replaced by an icon ghost box which you should position on the screen and plant by clicking the left mouse button.



Figure 5-26. A card icon ghost box requesting you to specify a position for the icon.

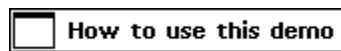


Figure 5-27. A card icon.

I recommend stacking your shrunk card on either the extreme left or right hand side of the screen. If you have Neat Icons turned on, from the **Background** menu, the system helps you stack the icons up neatly by aligning and evenly spacing them for you.

You expand icons by clicking the middle mouse button on the card, or selecting the **Expand** option from the right mouse button menu.

Closing Cards

There are two ways to close cards. You can position the cursor in the title bar of the card you want to close and, holding the right mouse button down, select the **Close** option. Or, you can again position the cursor in the title bar of the card you want to close. This time hold the left mouse button down, which will bring up the following menu.

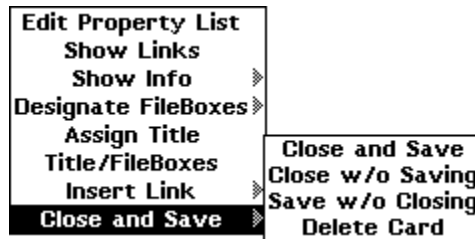


Figure 5-28. **Card Menu**

You can either select **Close and Save** or, if you have made changes that you don't want to save, slide off and select **Close w/o Saving** from the submenu. If you save changes that you later want to dispose of you can always choose **Abort** instead of **Close** from the Banner title-bar menu shown in Figure 5-21.

Creating Cards

To create new cards, you can click the left mouse button on the New Cards option on the notefile Banner, which will give you a Text card. To gain access to all the other card types, hold the middle mouse button down on the same **New Cards** option and select a card type from the **Card Types** menu.

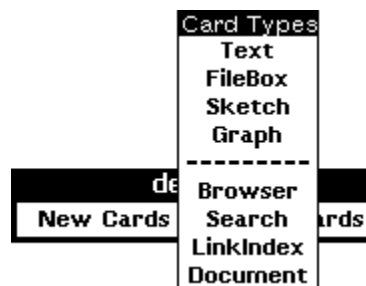


Figure 5-29. **Card Types Menu**

The system will show a ghost box for the card which you can position where you want.

Where To Go From Here

The preceding should get you started with NoteCards. Once you get the knack of some of the basics the system is quite easy to learn.

When you are ready to read the documentation you probably want to start with Chapter 7, The User Interface; the section in Chapter 8 dealing with the card menu, and browse through *A User's Guide to TEdit*. If that doesn't stop you, Chapter 10, User Cards and Chapter 11, System Cards are also good chapters to become familiar with. The chapter you probably want to slog all the way through is Chapter 8, Links. This chapter, more than any other, will give you a real feeling for what is going on in NoteCards.

[This page intentionally left blank]

6. BUILDING NOTECARDS STRUCTURES

NoteCards has been around for some time and people have developed different ways of using it. In this chapter we have recorded some of the different approaches people have taken towards using NoteCards in the hopes that these notes will help you discover the best way of using NoteCards for you.

If you have ideas about ways of using NoteCards which you would like to share with others, please send them to us.

Introduction

By its very design, NoteCards encourages you to use it in specific ways. However, some users have found that they make the best use of NoteCards by not following the intended path but by branching off and forcing NoteCards to conform to their own idea of what a hypertext system should look like.

A Standard View of Building NoteCards Structures

NoteCards was intended to support two parallel modes of organization, a hierarchial one, implemented with FileBoxes, and a relational one, implemented with links.

Here is one possible scenario:

1. Collect a body of material, and break it up into logical bits.
2. Enter the logical bits into Text, Sketch, Graph, and other card types, using a scanner or other electronic sources.
3. Build a hierarchical structure with FileBoxes, inserting the data cards in their appropriate places in the FileBox hierarchy.
4. Read through the document, building links between related ideas and constructing other non-hierarchical structures. Use different link types to keep these lines of thought apart.
5. Collect the different lines of thought in the material by using document cards to follow the different link types.

This is a stereotypical view of how you might generate a structure in NoteCards; however, it is not the only view.

A Non-Standard View of Building NoteCards Structures

FileBoxes and Hierarchies

One frequent comment from longtime users is that people have a tendency to use FileBoxes too soon. They try to organize their data into a structure before the best structure is apparent and wind up reorganizing their ideas, wasting much time and energy. These users suggest that you build your structures only after you have worked with your data for awhile.

Other users simply do not use FileBoxes at all or use them to build only very shallow structures. Note that you could use links directly to build the equivalent hierarchial structure as with FileBoxes.

Flat System

In the flat system, all cards are filed in the "Table of Contents", the "To Be Filed" or some user-specified FileBox card. These users may specify one or several start points for the reader, but aside from that the system is flat.

Almost Flat System

In this organization the cards are grouped into large bunches which can be based on source, topic or some other broad category. One FileBox is used for each source, topic, or whatever the divisions are. One user likens his use of FileBoxes to library shelves where each FileBox is a different shelf. He then uses Sketch cards as an organization and integration tool, to help him navigate through his system.

Link Type

The system is built with the idea that you will classify each link type so that later you can build documents or browsers based on link types. However, some users make all their link types "Unspecified."

One of the dangers with links is that new users tend to link everything to everything else and end up with spaghetti.

Multiple Users

NoteCards was not designed with collaboration in mind. One way users have gotten around the problem of tracking what each user has done is to assign each collaborator a different font. Each contributor's work can then be recognized by font face.

Cards and Information Chunk Size

Because cards come up on the screen a particular size, able to display a small amount of information, the perception grows among new users that the information chunks have to be small. It is important to recognize that cards can hold any amount of information, and that they can be scrolled or resized to display all the information they hold.

There are trade offs, however. If you work with large cards, the size of your notefile grows much more rapidly. This is not serious. It only means that you need to compact your notefile more frequently than people who work with smaller information chunks do.

A second trade off is that links always point to the beginning of a card never into the body of a card. If your cards are very large, it may not be immediately clear what the relationship is between the source card and the destination card.

Types of Structures Built with NoteCards

One user working on a design project used a notefile structured with design goals as roots moving down to the component level at leaf nodes. This structure allowed him to see the design history, subgoals, and goals which were retracted.

Other users use NoteCards to maintain indices of where to find sources of information. This is fairly common. Users frequently implement on-line rolodexes, customer-support or customer-tracking notefiles, and other types of text databases, such as project tracking.

The most common use is in writing large reports. NoteCards is used to collect, organize, and cross-reference information until the writer has a firm grasp of the material and how various pieces of it interact.

Specializing NoteCards

It is possible to specialize NoteCards to specific tasks.

One user has created an interface to the Lexis legal database. To do this he created a new text-card type which receives retrieved data from Lexis. He is also working on an interface to videodisk and other sources.

At Xerox PARC there is a group working on an Instructional Design Environment for designing, developing, and presenting instruction. These people are extensively modifying the NoteCards environment

If you have a need for a particular extension to NoteCards there is a programmers interface which you or your company's developers can use to extend the NoteCards environment, or Venue can provide you with specialized extensions to NoteCards.

[This page intentionally left blank]

7. USER INTERFACE

This chapter covers the basics of using the window system on which NoteCards is built. Read this chapter if you are a new user of the system or if you need to understand how to manipulate windows, icons, and use the functions on the background menu. For a more detailed description of how to use the mouse and keyboard, see Chapter 5.

This chapter explains:

- Where to find system menus
- What these menus do

Window Menu

The NoteCards window system allows you to interactively manipulate the windows on the screen (move them around, change their shape, etc.) by selecting various operations from the **Window** menu (Figure 7-1).

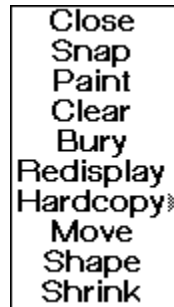


Figure 7-1. **Window** Menu

When you press and hold down the right mouse button in a window's title bar, as shown in Figure 7-2, that window comes to the front and a menu of window operations appears. In some cases, you can hold down the right mouse button anywhere in the window to access the **Window** menu.

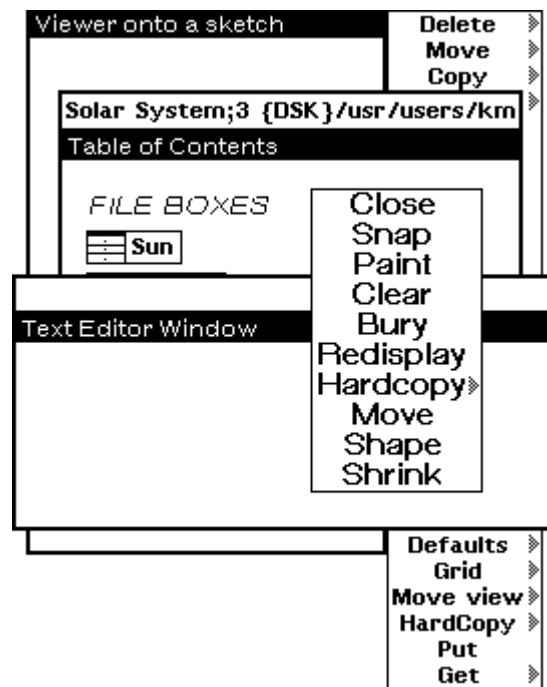


Figure 7-2. Window Menu from a TEdit Window Title Bar

The ten options available on the Window Menu are explained below.

Close

Closes a window, i.e., removes it from the screen. If you have modified the contents of the window, the system may ask you to confirm, by pressing the left mouse button, that you want to close the window and lose your changes (Figure 7-3). Press the right or middle mouse button to cancel the close operation.

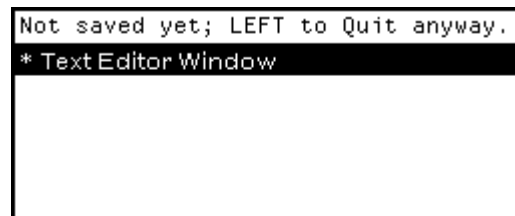


Figure 7-3. TEdit Window Requesting Confirmation of Close Command

Snap

Prompts for a region on the screen and makes a new window whose image is a snapshot of the image currently in that region (Figure 7-4). This is useful for saving some particularly choice image before the window image changes.



Figure 7-4. Mouse Cursor Prompting Sweep-out Window Size and Shape

Most of the images presented as figures in this manual are screen snaps.

When making a snap, it is possible to change corners and change the snap shape from a different side of the ghost frame. In the middle of a snap process, while you still have the left mouse button down, simultaneously press and hold down the right mouse button.

The forceps cursor () appears. Move this cursor to the corner of the ghost frame you want to move, and, still holding the left mouse button down, release the right mouse button. You can now adjust the snap from the corner you just selected.

Paint

Switches to a mode in which the cursor can be used like a paint brush to draw within a window. This can be useful for making notes on a window or touching up snaps. While the left key is down, pixels are blackened. While the middle key is down, they are erased. The right button pops up a command menu (Figure 7-5) that allows you to change the brush mode, shade, shape, and size, and to quit the paint utility. These five options are described below.



Figure 7-5. **Paint Command Menu**

SetMode Brings up the menu shown in Figure 7-6.



Figure 7-6. **Mode Menu for Paint**

Changing the paint mode changes the way the bits in the brush combine with the existing bits in the snap. Figure 7-7 shows a sample bit map pattern from a snap. The horizontal bars are the pattern in the example brush. Figure 7-8 illustrates combinations of the snap and brush patterns shown in Figure 7-7.

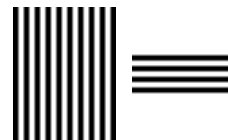


Figure 7-7. **Sample Bitmap Pattern of Vertical Bars**

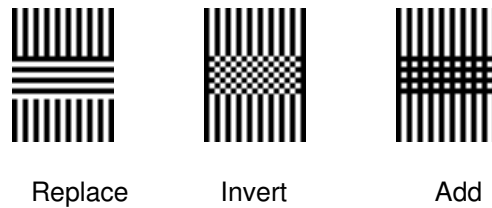


Figure 7-8. Possible Combinations of Snap and Brush Patterns

SetShade Pops up the **Choose Shade** Menu (Figure 7-9) from which you can select one of a pre-existing set of shades or specify your own 4x4 shade. New shades are added to the **Choose Shade** Menu.

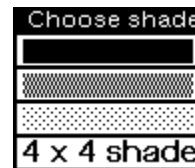


Figure 7-9. **Choose Shade** Menu

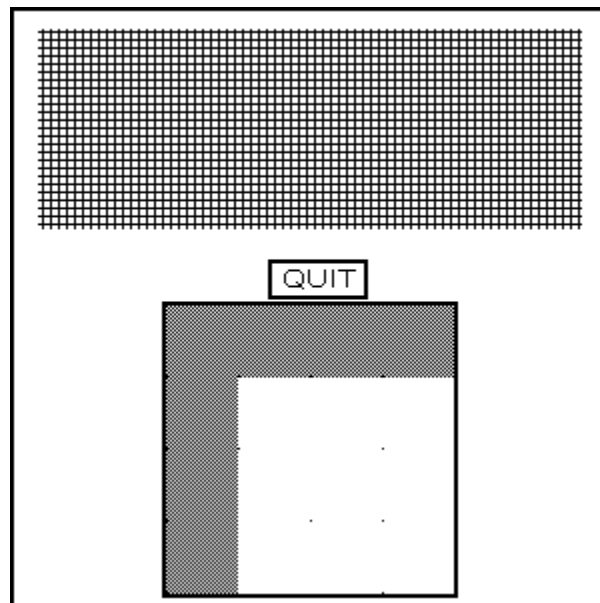


Figure 7-10. 4x4 Shade Tool

Set Shape Pops up the **Shape** Menu (Figure 7-11), which allows you to set the shape of the mouse cursor.



Figure 7-11. **Shape** Menu

Set Size Brings up the **Size** Menu (Figure 7-12), which sets the cursor brush size.

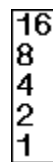


Figure 7-12. **Size** Menu

Quit Quits the paint program.

Clear Clears the window. Sets all pixels in the window to white.

Bury Places the window under all other windows which it covers or overlaps, thereby exposing any windows that it was hiding.

Redisplay Clears the window and rewrites the original window contents to the window.

Hardcopy Sends the contents of the window to the printer. If the window is associated with a text, sketch, or graph editor, the editor's contents are printed. In the case of TEdit and Sketch, this means that the associated TEdit file, if there is one, is printed. If the window is not associated with an editor, a bit-map image of the window is sent to the printer.

To save the image in a PostScript, InterPress, or Press formatted file, or to send it to a non-default printer, use the the **Hardcopy** command submenu. When the mouse is moved off to the right of the **Hardcopy** menu item, a second pop-up menu appears giving the choices **To a file** or **To a printer**.

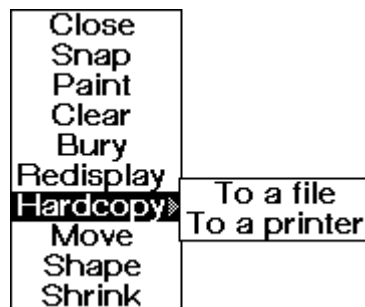


Figure 7-13. **Hardcopy** Submenu

If you select **To a file**, you are prompted for a file name, and the format of the file (PostScript or Interpress) from the **File type?** Menu and the contents of the window is stored in that file, formatted for the printer type you specified.

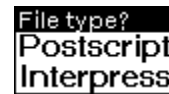


Figure 7-14. **File type?** Menu

The system provides the suffixes for these files. PostScript files have a `ps` suffix. InterPress files have an `ip` suffix. For example, given a file name of `Letter`, the system stores a file with the name `Letter.ps` for a PostScript formatted file and `Letter.ip` for an InterPress formatted file. If you change the suffix for a formatted file from `ps` or `ip` to some other name, the system prompts you for the file type when you try to print the file (Figure 7-15). For a formatted file, select **BINARY**. For an unformatted file, select **TEXT**. Use the **Hardcopy** option on the FileBrowser Menu to print a formatted file.



Figure 7-15. **File Type?** Menu

If you select **To a printer**, the **Which Printer?** Menu is displayed (Figure 7-16), prompting you to select a printer from the list of known printers, or to type the name of another printer.



Figure 7-16. **Which printer?** Menu

The topmost printer in the **Which printer?** Menu is the default printer. If the printer selected is not the topmost (default) printer on the list, you are asked whether to make the selected printer the new default printer (Figure 7-17).

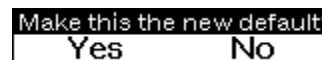


Figure 7-17. **Make this the new default?** Menu

Unless you are using only fully specified Interpress printer names or you have set the `DEFAULTPRINTERTYPE` variable from your initialization file, it is not advisable to use the **Other...** option on the **Which printer?** Menu. This option does not allow you to specify the printer type.

Move

Moves the window to a new screen location. When you select **Move** you are presented with a ghost frame which is the size and shape of the original of the window you are moving (Figure 7-18). Move the mouse to position the ghost frame where you want the window to appear and then plant the window by clicking the left mouse button.

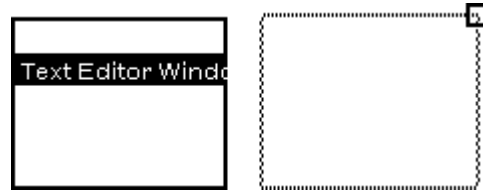


Figure 7-18. TEdit Window and its Ghost Frame

Shape

Allows you to specify a new size and shape for an existing window. You can use either the left or middle mouse buttons to shape a window.

The left mouse button allows you to change the size, shape, and location of a window. Once you have selected **Shape** from the **Window** Menu, position the cursor where you want the upper left corner of the window to be located, press and hold down the left mouse button, sweep out a new region for the window, and release the mouse button when the window has the size and shape you want.

The middle mouse button allows you to change the size and shape of a window by moving just one corner of the window. Select **Shape** from the **Window** Menu and position the mouse cursor at the corner you want to move. Press and hold down the middle mouse button, and move the mouse cursor to the location at which you want that corner of the window to be. A ghost frame appears to indicate the new size and shape of the window.

When reshaping a window with either the left or middle mouse button it is possible to change corners and change the window shape from a different side of the window. In the middle of a shape process, while you still have the left or middle mouse button down, simultaneously press and hold the right mouse button down.

When you do this, the forceps cursor () appears. Move this cursor to the window corner you want to move, and, still holding the left or middle mouse button down, release the right mouse button. You can now reshape the window from the corner you just selected.

If you have too many windows on the screen and you are having problems accessing them, you probably want to try shrinking some windows.

Shrink

Removes the window from the screen and brings up its icon. The window can be restored by clicking the middle

mouse button inside the window's icon, or by selecting **Expand** from the right button icon menu. Figures 7-19 through 7-23 illustrate some of the shrink icons.

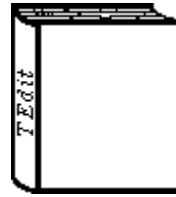


Figure 7-19. TEdit-window shrink icons look like books.



Figure 7-20. Sketch-window shrink icons look like sketch pads



Figure 7-21. FileBrowser-window shrink icons look like filing cabinet drawers

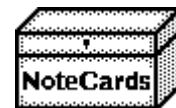


Figure 7-22. NoteCards MenuBox shrink icon looks like a recipe file box

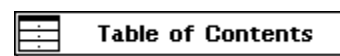


Figure 7-23. Notecard shrink icons look like link icons with an extra border

Icon Menu

Icons are a variant of windows and have a menu (Figure 7-24) similar in form and function to the window menu.

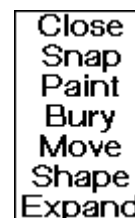


Figure 7-24. Icon Menu

For the options Close, Snap, Paint, Bury, Move, and Shape see the discussion above under the "Window Menu" heading. The Expand option is described below.

Expand Restores the window associated with the icon accessed and removes the icon image from the screen. You can also expand an icon by clicking the middle mouse button inside the icon.

Background Menu

The **Background** Menu (Figure 7-25) appears when the right button is pressed while the cursor is not in any window.

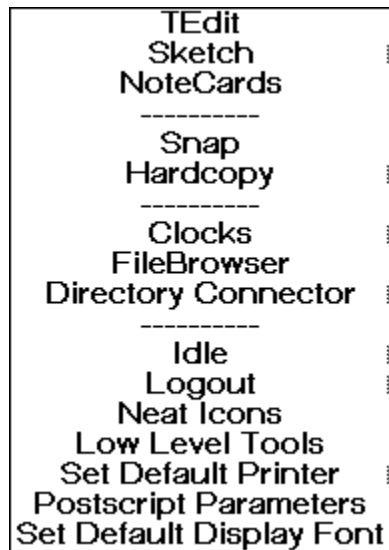


Figure 7-25. **Background** Menu

The **Background** Menu options are described below.

TEdit Opens a new TEdit window and starts a new TEdit session.

Sketch Opens a new Sketch window and starts a new Sketch session.

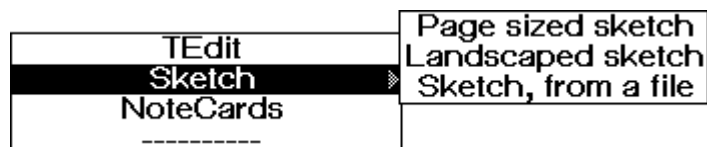


Figure 7-26. **Sketch** Submenu Options

The three Sketch Submenu options (Figure 7-26) are described below.

Page sized sketch Allows you to start a sketch which will fit exactly on an 8 1/2 x 11 inch sheet of paper when held in the normal vertical, or portrait, position.

Landscaped sketch	Allows you to start a sketch which will fit exactly on an 8 1/2 x 11 inch sheet of paper when held in a horizontal, or landscape, position.
Sketch, from a file	Allows you to start a pre-existing sketch which is stored in a file.

NoteCards Opens the NoteCards MenuBox Icon, which is the interface for a NoteCards session.

Snap The same as the **Window** Menu command Snap described above.

Also, TEdit, Sketch, and NoteCards allow information to be shift-inserted at the current cursor position by selecting an area of the screen with the SHIFT key held down. To shift-insert the bitmap of a snap into an editor, position the cursor where you want the image to appear, hold the SHIFT key down, press and hold down the right mouse button in the background, and select **Snap** from the single item menu which appears. Finally, sweep out the area for the snap and release the mouse button and the SHIFT key. The snap appears in the editor where the cursor was positioned. Sometimes it is necessary to scroll or redisplay the window for the snap to appear.

Hardcopy Prompts for a region on the screen, and sends the bitmap image to the default printer. Note that the region can cross window boundaries.

Like the Window Menu **Hardcopy** command discussed above, you can print to a file or specify a different printer than the default by using the **Hardcopy** submenu. See the discussion of the Hardcopy command under the "Window Menu" section above, for more detail.

Clocks Starts an analog clock and places it on the screen. The **Clocks** submenu (Figure 7-27) gives you access to a digital clock.



Figure 7-27. Clocks Option on **Background** Menu

For a complete discussion of this menu option, see Chapter 15.

FileBrowser Opens a FileBrowser window which prompts you for a directory to browse.

For a complete discussion of this menu option, see Chapter 14.

Directory Connector

Opens a window which displays your currently connected directory, and allows you to change directories.

For a complete discussion of this menu option, see Chapter 15.

Idle

Enters idle mode, which blacks out the display screen to save the phosphor. Exit the idle mod by pressing any key on the keyboard or mouse. This menu command has subitems that allow you to interactively set idle options to erase the password cache (for security), to request a password before exiting idle mode, to change the timeout before idle mode is entered automatically, etc.

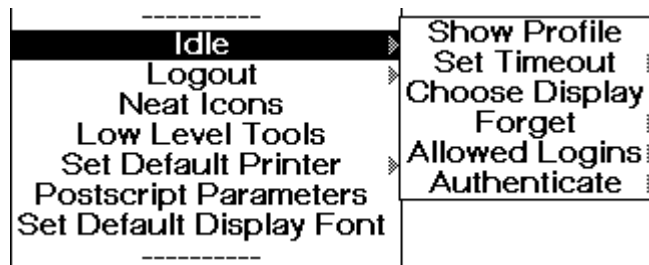


Figure 7-28. **Idle** Submenu

If either shift key is pressed while NoteCards is in idle mode, the current user name and the amount of time spent idling are displayed in the prompt window. This information appears as long as the shift key is held down.

The six Idle Submenu options are described below.

Show Profile

Displays the current idle profile in the system prompt window.

Set Timeout

Sets the amount of time that the machine will wait for a key stroke or mouse click before automatically going into idle mode.

When you select this option, the system brings up a number pad on which you can enter the new idle time out. An idle time out of zero sets the machine so that it will never go into idle mode.

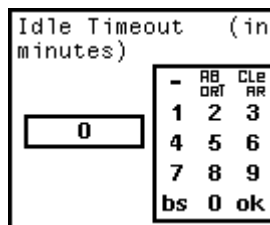


Figure 7-29. **Number Pad** Menu Requesting New Idle Timeout

The submenu option **Never** sets the machine so that it never goes into idle mode. We counsel against using this option as it can result in the phosphor on your screen being damaged.

Choose Display Allows you to select what will be displayed on the blacked out screen when the machine is in idle mode. Selecting this option brings up the two-choice menu shown in Figure 7-30 and described below.



Figure 7-30. **Idle Display** Menu

Bouncing Box Chooses the Venue logo as the idle display.



Figure 7-31. Bouncing Box Venue Logo

Bouncing Username Chooses the your username as the item to bounce around on the screen while in idle mode.

Forget If set to Don't (or NIL), your password is not erased when idle mode is entered. Default is Do (or T(rue)) erase password. The initialization file distributed with the NoteCards system resets **Forget** to Don't.



Figure 7-32. Forget Submenu

Note: If the password is erased, any programs left running when idle mode is entered will fail if they try doing anything requiring passwords, such as accessing file servers.

Allowed Logins Determines who can exit idle mode.

The four **Allowed Logins** submenu options are described below.

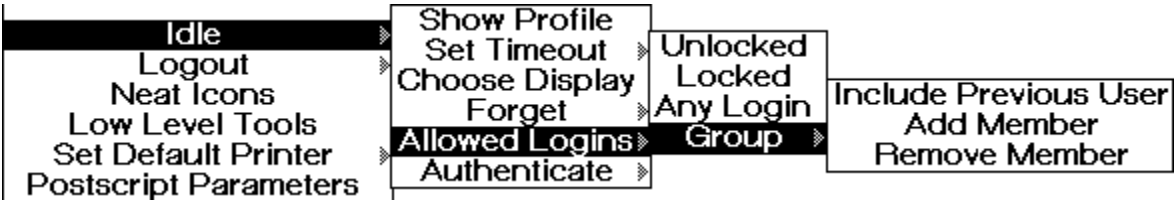


Figure 7-33. **Allowed Logins** Submenus

Unlocked	Sets the system such that login is not required to exit idle mode. Initialization file value is NIL.
Locked	Lets only the previous user exit idle mode. Initialization file value is (T) .
Any Login	Require login, but let anyone exit idle mode. Login overwrites the previous user's user name and password each time idle mode is exited. Initialization file value is (*) .
Group	Allow any members of a specified group to exit idle mode. The Group Submenu options (Figure 7-33) allow you to add and delete group members.
Authenticate	The value of this property determines what mechanism the system uses to check passwords.

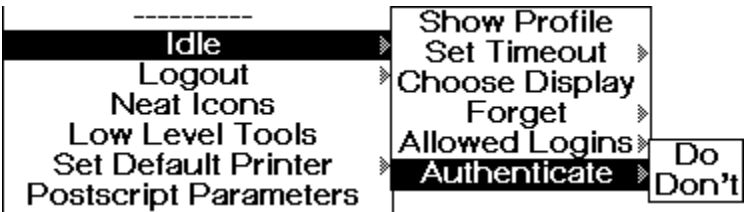


Figure 7-34. **Authenticate** Submenu

If the value of this property is `Do` or `T` (true), the Xerox network system protocol is used to do the authentication.

If the property value is `UNIX`, the Sun Operating System is used to do the authentication. This option is set in the standard initialization file. See Appendix C for a complete discussion.

If the value of this property is `Don't` or `NIL`, the password is not checked. Any password is accepted.

Logout	Saves the NoteCards virtual memory working image to a file and returns the system to UNIX. This is the normal way to return to UNIX.
---------------	--

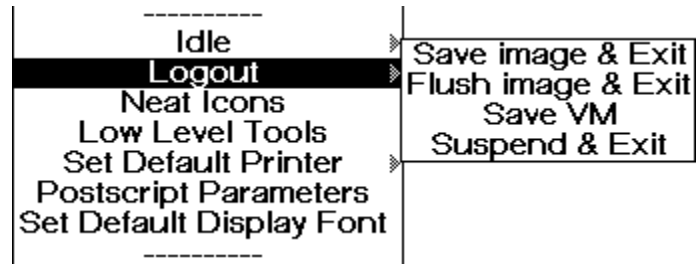
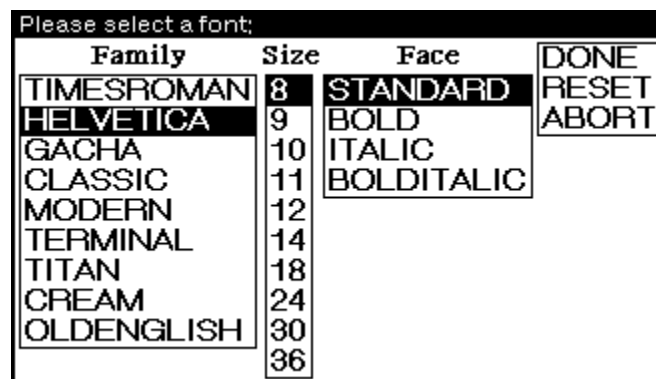


Figure 7-35. **Logout** Option and its Background Menu Submenu

Save image & Exit	The same as Logout .
Flush image & Exit	Returns the system to UNIX without saving the working image. Use this option when you have saved all your work to files and you do not care if the system comes up in the same state it was in when you left it.
Save VM	Saves the NoteCards virtual memory working image to a file but does not return the system to UNIX. Use this option frequently if you are concerned about power failures or other problems which may cause your machine to crash.
Suspend & Exit	Suspends the NoteCards session and returns control to UNIX. You restart the NoteCards session by entering the command fg (foreground), at the UNIX prompt.
Low Level Tools	This menu option provides people extending the NoteCards system access to the Lisp executives, the process status window, and allows them to set the Show TEdit Props flag. The NoteCards user should never access these options except under the instruction of a developer or support person.
Set Default Printer	Allows you to specify which printers you want to print to and which one is the default. See Chapter 16 for a complete discussion of this menu option.
Postscript Parameters	Allows you to control the way PostScript documents are printed. See Chapter 16 for a complete discussion of this menu option.
Set Default Display Font	Brings up a series of menus (starting with the Choose Font Profile to Update Menu shown in Figure 7-36) which allow you to specify what fonts will be used on the screen. This is a particularly useful option if you are using a high resolution screen and want to increase font size or for people who are vision impaired.

Figure 7-36. **Choose Font Profile to Update** Menu

To change a display font, select the **Set Default Display Font** background menu option and choose the font you want to change from the **Choose Font Profile to Update** Menu. Each of the font options is discussed below. Once you have selected the font you want to modify, the **Please select a font:** Menu appears. Select the font family, size, and face you want; then select **DONE**. **RESET** makes the menu display the current setting for the default display font you are modifying. **ABORT** terminates this operation and closed the **Please select a font:** Menu.

Figure 7-37. **Please select a font:** Menu

If you choose a font which does not exist, or the system is unable to find it, the old font specification remains unchanged.

If the system is unable to find a font which you know to exist, check the setting of the display font directory list. This is done with the **Show Font Directories** subsubmenu option off the **Set Default Printer** background menu option. See Chapter 16 for a complete description of how to use this option.

The five options available on the **Choose Font Profile to Update** Menu are described below.

FileBrowser Display Font

The font in which the information in the main display window is printed, initially 10-point Gacha.

FileBrowser Prompt Font

The font in which FileBrowser prompt messages are printed. Initially 8-point Gacha. Changing this value only affects new FileBrowsers created from the background menu. Existing FileBrowsers are unaffected.

Prompt Window Font

The font used to print information in the black system-prompt window.

Default Font	<p>The starting font used in TEdit and Sketch windows as well as all other windows which do not have an explicitly specified font associated with them.</p> <p>The default font for NoteCards cards is specified from the NoteCards System Parameters Menu. See Chapter 13 for a complete discussion of how to specify this default font.</p>
Menu Font	<p>The font used in all system menus. The menu font is used in the window, icon, background, and other menus.</p>

[This page intentionally left blank]

8. LINKS

A link is a connection between a source card and a destination card. A link icon (see Figure 8-1) represents the link.



Figure 8-1. Link Icon Image

Click on the link icon to display the destination card. Links are the glue which holds together the relationships and conceptual structures represented in a notefile.

This chapter explains:

- What links and link types are
- How links are accessed and used
- How to create links
- How to delete and undelete links
- How to tailor links to your taste
- What cross-file links are

Links and Link Types

Link types specify the nature of the relationship between the source and destination cards. Some typical link types are Question, Answer, and Explanation.

Two groups of link types exist in the NoteCards system: one group is system-reserved, the other is user-specified. Some of the system-reserved links support the FileBox hierarchy that NoteCards helps you to create. The user-specified links add a relational dimension to information structuring. Here you can link cards together to form networks. These networks allow you to represent the interconnections between various ideas or pieces of information, independent of any categorization into topic areas. In contrast, the set of FileBoxes forms a strict hierarchy typically representing a breakdown of information into subtopics or subcategories. By strict hierarchy, we mean that you are not allowed to create circular lists of FileBoxes. Circular lists of other card types are allowed, however.

System-Reserved Link Types

SubBox	A link pointing to a FileBox from another FileBox.
FiledCard	A link pointing from a FileBox card to any non-FileBox card. Text, Sketch, Graph, Browser, Search, LinkIndex, and Document cards are non-FileBox cards.
BrowserContents	A link pointing to a card or box from a Browser card. This type of link is traversed when you bring up any of the cards or boxes represented as link icons in a Browser card.

ListContents	A link pointing to a card or box from a Search card, enabling you to bring up any of the cards or boxes found during the search process.
LinkIndexBackPtr	A link pointing to a card or box from a LinkIndex card. This type of link is built when you specify that a LinkIndex card should have back pointers to the cards and boxes being indexed, i.e., link icons should be included in the LinkIndex card.
DocBackPtr	A link pointing to a card or box from a Document card. This type of link is built when you specify that a Document card should include back pointers to each card and/or box used to build the document.

User-Specified Link Types

User-specified links indicate the logical relationships between cards. The system comes with three predefined user-specified link types: Comment, See, and Unspecified. However, the system allows you to create new link types as you need them. Typical link types users create include Explanation, Example, Question, Answer, Next, and Source. Choose your link types carefully as they determine the lines along which you will be able to extract information from your notefiles with Document cards.

Randy Trigg, one of the developers of the NoteCards prototype at Xerox PARC, proposed a long list of standard link types for users to follow in categorizing the relationships between their notes. A partial list is included here to give you some idea of the possibilities.

- Citation
 - C-source
 - C-credit
 - C-leads
- Argument
 - A-deduction
 - A-induction
 - A-analogy
 - A-intuition
- Background/Future
- Refutation/Support
- Methodology/Data
- Solution
- Continuation
- Correction/Update
- Simplification/Complication
- Explanation
- Summarization/Detail
- Alternate-view
- Rewrite
- Generalization/Specification
- Abstraction/Example
- Formalization/Application

Note that some long-time users of the system use only one link type. NoteCards gives you a great deal of flexibility. Just because some functionality is there does not mean you should feel obligated to use it if it is not helpful to your task.

Link Directions and Link Ends

Links have a single start point and a single end point. However, whenever you create a link from one card to another, the system automatically builds a backwards link so that it is possible to traverse a linked sequence of cards in either direction. The result is that all cards are linked bidirectionally.

A link's start point is either the card itself or somewhere in the card's contents. In the case of a text card, this would be somewhere within the body of the card's text. A link's end point is always another card. Thus, there are three distinct link categories. There are also a few terms which refer to different groupings of these categories.

Link Categories

To-Links	Links whose start point is in a card's contents and whose end point is another card. These links are represented as link icons in a card's body.
Global To-Links	Links whose start point is a card itself and whose end point is another card. These links are not represented by link icons in a card's body. To see and access these link icons you must select the Show Links option from a card's left-button title-bar menu.
From-Links	For every to-link and global to-link there is a corresponding from-link which points from the destination card to the source card. All from-links are global.

Other Link Terminology

Local Links	Links whose start point is somewhere in the contents of a card.
Global Links	Links whose start point is the card itself and not some point in the card's contents. These links are not represented by link icons in a card's body. To see and access these link icons, you must choose the Show Links option from a card's left-button title-bar menu.
Forward Links	Include both to-links and global to-links. This term is used when setting Browser and LinkIndex card parameters.
Backward Links	The same as from-links. This term is used when setting Browser and LinkIndex card parameters.
Back Links	System-generated to-links. These links point from a Document or LinkIndex card <i>back</i> to the source cards from which the Document or LinkIndex was built. Hence the name back link.

Link Types vs. Card Types

Link types and card types are frequently confused but they are totally independent of each other.

Link types classify the assigned relationship between two cards. Link types frequently include Explanation, Question, Answer, and Citation.

A card's type specifies the functional capabilities of that card and is associated with a text or graphics editor. The eight card types are Text, FileBox, Search, LinkIndex, Document, Sketch, Graph, and Browser. The Text, FileBox, Search, LinkIndex, and Document cards are all based on the text editor TEdit. The Sketch card is based on the Sketch editor and the Graph, and Browser cards are based on the graph editor Grapher.

Link Access and Use

Link Icon Functionality

A link icon is an active region with an associated menu and four functions. The most frequently used functionality is to click the left mouse button in the body of the link icon to traverse the link to the destination card. The middle mouse button brings up a menu which allows you to traverse the link, change the link type, change the card title and change the link display mode.

Active Regions of Link Icons

As illustrated in Figure 8-2, the entire link icon (except for 10 pixels at the left edge and 10 pixels at the right edge) responds to mouse clicks by carrying out a NoteCards action.

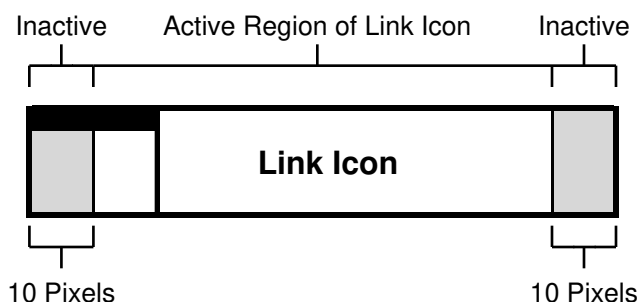


Figure 8-2. Link Icon Active and Inactive Regions

Clicking in the "inactive" 10-pixel strips at the left and right edges of the link icon selects the link icon, in text-based cards (Text, FileBox, Search, LinkIndex, and Document cards), for delete, copy, and move operations, without traversing the link. This region does not exist on sketch- and graph-based cards (Sketch, Graph, and Browser cards).

The inactive area can be somewhat difficult to access without touching the active region of the card. One way to access it is to hold the left or middle mouse button down and slide into the link icon's inactive region. You have selected the link icon when it appears underlined, as shown in Figure 8-3.

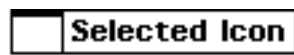


Figure 8-3. Selected Icon

If you go too far and the card redisplay in reverse video, simply slide the mouse cursor out to the edge of the link icon and watch for it to underline again. Once the icon appears underlined release the mouse button.

If a link icon is so narrow that its active region would be less than 10 pixels wide, the active region is set to the middle 50% of the link icon width, with 25% at left and 25% at the right being the inactive strips. If the link icon is less than 10 pixels in width altogether, then the link icon will have no inactive strips. Its entire area will be active.

Mouse-Button Actions in Link Icons

The exact response to a mouse click in the active region of a link icon varies depending on a number of conditions. For more general mouse button behaviors, see Chapter 7 as well as the *User's Guide to TEdit* and the *User's Guide to Sketch*.

Left Button	Brings up the destination card represented by the link icon.
Middle Button	Brings up the link icon menu which allows you to change the link type, title, and display mode.
Right Button	Following a left- or middle-button click elsewhere in a TEdit-based card, selects a region of the window for moving, copying, or deletion which can include link icons.

COPY/SHIFT or MOVE/SHIFT-CONTROL key plus mouse-button operations are three-step operations.

1. Specify the destination window by clicking the left mouse button to position the flashing cursor in the window.
2. Hold the appropriate keyboard key down.
3. While still holding the key down, use the mouse to select the piece of text/graphics to be operated on.

Once you have completed your selection, release the keyboard key and what you have selected will be copied or moved to the destination window.

Link icons should only be copied or moved to NoteCards windows. TEdit and Sketch, when used outside of NoteCards text-based and sketch-based cards, will not understand link icons.

Copy/Shift key + Left Button

This operation is frequently referred to as a shift-select, occasionally as a copy-select. This operation copies a link icon, or link-icon name and link, to another card.

If the flashing cursor is in the body of a card, holding the SHIFT or COPY key down and clicking the left mouse button on a link icon copies the link icon and the link it represents to the position of the caret cursor in the destination card.

If the flashing cursor is a crosshairs and is in a **Selecting cards to file** prompt window, holding the SHIFT or COPY key down and clicking the left mouse button on a link icon will copy the link icon name and link to the select-card prompt window. This operation is used when you are inserting links in cards or filing cards in a FileBox card.



Figure 8-4. Select Card Prompt Window

Copy/Shift key+ Middle Button

Same as left button.

Copy/Shift key + Right Button

Holding the COPY or SHIFT key down and pressing the right mouse button following a left- or middle-button click elsewhere in a text-based card, selects a region of the window for copying which can include link icons.

Move/Shift-Control + Left Button

Moves a link and link icon from one card to another card by deleting the link in the first card and inserting it into the second. To move a link, position the caret cursor in the destination window where you want the link to appear and, holding the MOVE key down or holding the CONTROL and SHIFT keys down simultaneously, select the link icon you want to move with the left mouse button. When you release the key or keys you are holding down, the link icon will move to its new position.

Move/Shift-Control + Middle Button

Same as left button.

Move/Shift-Control + Right Button

Holding the MOVE key or SHIFT and CONTROL keys down and pressing the right mouse button following a left- or middle-button click elsewhere in a text-based card, selects a region of the window to be moved which can include link icons.

CONTROL key plus mouse button operations are two-step operations. First, hold the CONTROL key down. Second, while still holding the key down, use the mouse to select the piece of text/graphics to be deleted. Once you have completed your selection, release the CONTROL key and what you have selected will be deleted.

Control key+ Left Button

Deletes a link and link icon from a card. Holding the CONTROL key down and pressing the left mouse button on a link icon will select the link icon for deletion. This is shown by displaying the link icon in reverse video, as illustrated in Figure 8-5.



Figure 8-5. Selected Link Icon

The link icon and the link it represents are deleted when you release the CONTROL key. You can cancel this operation before you release the CONTROL key by clicking elsewhere in the window to deselect the link icon.

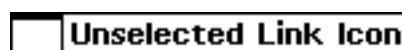


Figure 8-6. Unselected Link Icon

Press the UNDO key to undo this operation after you have released the control key.

Control key + Middle Button

Same as left button.

Control key + Right Button

Holding the CONTROL key down and pressing the right mouse button following a left- or middle-button click elsewhere in a text-based card, selects a region of the window for deletion which can include link icons.

It is also possible to just backspace over a link icon or to select it and then press the DELETE key. Note that on some Sun keyboards, the backspace key has the word "Delete" printed on it and that the NoteCards DELETE key is one of the function keys, usually the key labeled "L10."

Viewing Local and Global Links

All links are stored in three places:

- As a to-link on a source card
- As a from-link on a destination card
- Either for local to-links, in the card's contents as a link icon, or for global to-links, in the card's global links list

Only the local to-links, the links in a card's contents, are immediately visible.

To see all of the links to and from a card, hold the left mouse button down in the card's title bar and select the **Show Links** option (see Figure 8-7).

Figure 8-7. Card Menu with **Show Links** Option Selected

This opens a **List of Links** Window (see Figure 8-8). This window always opens with the same initial height. If the list of links is longer than three or four, you will need to scroll or reshape the window to see the rest of the links. The link icons in this window have the standard left-button functionality, you can traverse links, but no middle-button functionality. Links can also be copied, moved, and deleted from this window.

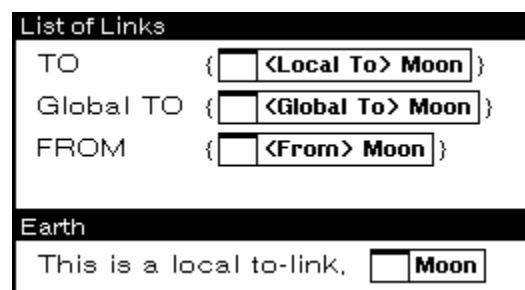


Figure 8-8. List of Links Window with Links to and From Earth and Moon Cards

The headings to the left of the links are active but have no operations associated with them.

You can close the List of Links window by holding down the left or middle mouse button in the window title bar and selecting **Quit** from the single-option menu (shown in Figure 8-9), or by pressing the right mouse button in the title bar and selecting the **Close** option from the standard window menu.



Figure 8-9. List of Links Left Button Menu

Unfiled and Lost Cards

All cards, including FileBox cards, must be filed in at least one FileBox. This assures that you will have a pointer to your information and helps to remind you to hierarchically organize your information in addition to specifying other types of relationships. When you close a card without linking it to a FileBox card, the closed card is placed in the To Be Filed FileBox. You can access this FileBox card by holding the middle mouse button down on the notefile-banner **Special Cards** menu option and selecting the **To Be Filed** option from the menu which appears (see Figure 8-10).

Figure 8-10. **To Be Filed** Option on Special Cards Menu

From this FileBox all "unfiled" cards are accessible. It is possible to force yourself to file cards in a FileBox by setting the **Force Filing Edit Parameter**. See Chapter 13.

You can remove all FileBox links to a card by accident or by design. When you do this, the following message is displayed in the System Prompt Window. "You have just unfiled *card name* from its last filebox. It is being filed in the Orphan FileBox." You can access this FileBox card by holding the middle mouse button down on the notefile-banner **Special Cards** menu option and selecting the **Orphans** option from the menu which appears.

Finally, you can delete card links from the "Orphans" FileBox. When you do this the following message is displayed in the card's prompt window. "You have just deleted the last filing link to *card name*. The Search operation can be used to find it." At this point, if there are no other cards with links to this card, it becomes a lost card and there is no simple way to access the card even though it still exists. As the message says, though, you can retrieve lost cards by building new links to them with a Search card. However, unless you remember some fragment of the card's name, using a Search card to find lost cards can be a time consuming task in a large notefile.

Creating Links

The links you create always run from a source card to a destination card, as illustrated in Figure 8-11. References to the source card refer to the card with the link icon.

References to the destination card refer to the card which opens, or flashes if already open, when you click on the link icon.



Figure 8-11. Link Icon Pointing to its Destination Card

Creating Links in Text Cards

The procedures for creating links are basically the same for all cards. We will use Text cards as the model, and then discuss the differences between Text cards and other cards.

Insert Link

Use the **Insert Link** command to create a link originating in the source card's contents and terminating at another card. This is a multi-step process. First, you place the caret cursor in the card at the position where you want the link to appear by clicking in the source card's window. If the type-in process belongs to another card or window this usually takes two mouse clicks, one to make the card the active card and one to position the mouse cursor. Once you have established the location for the link icon in the source card, hold the left mouse button down in the source card's title bar and select the **Insert Link** option.

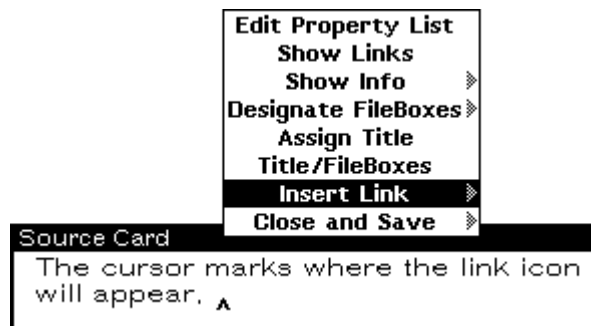


Figure 8-12. Card Menu with **Insert Link** Option Selected

At this point the **Link Type** menu opens, asking you to specify the link type.



Figure 8-13. **Link Type** Menu with **--New Link Type--** Option Selected

Choosing appropriate link types is important as they determine the ways you extract information from your notefiles. For more information on link types see the User-Specified Link Types section in this chapter and the on Document Cards section in Chapter 12. You can create new link types as you need them by selecting the **--New Link Type--** option and typing the name of the new link type into the prompt window which will appear above the source card (see Figure 8-14). When you press RETURN at this prompt widow, without defining a new link type, the system assigns the value "Unspecified" to the link type.

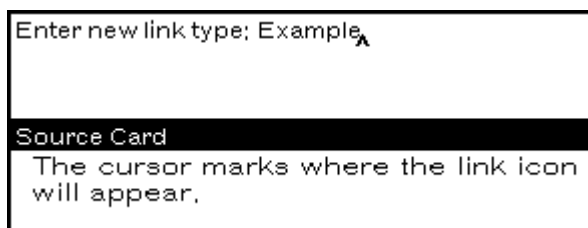
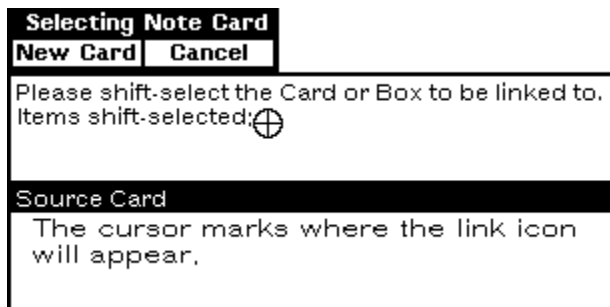


Figure 8-14. Note card Prompt Window Requesting New Link Type

After you have selected an existing link type or defined a new one, the system opens a **Selecting Note Card** window and prompts you to shift-select the card UID from the destination card (see Figure 8-15).

Figure 8-15. **Selecting Note Card** Window Prompting Shift-Selection of Card ID from Destination Card's Title Bar

A card UID is shift-selectable from three locations:



When shift-selecting the card UID, there are two distinct behaviors. From the link icon, it is necessary to release the SHIFT or COPY key after each selection. This is the expected behavior when shift-selecting anything from an edit window. From the title bar and shrink icon, it is not necessary to release the SHIFT key. The transfer happens immediately.

What you see transferred is the destination card's title, but NoteCards also maintains an identifier for each card which is unique across all cards and notefiles. This is what you are really shift-selecting from one card to another. And this is also the reason why it is not possible to shift-select or type text into this window.

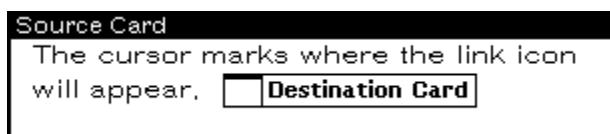


Figure 8-16. Link Icon to Destination Card in a Source Card

At this point, the new link icon appears in the source card. From now on, when you click on this link icon, it will open the destination card, or flash the destination card if it is already open.

Selecting **New Card** from the menu on top of the **Selecting Note Card** prompt window brings up a menu allowing you to create a new card, of any type, which will be the destination card.

Selecting **Cancel** from this menu cancels the entire **Insert Link** operation.

At any point in this process, it is possible to transfer the type-in process from a prompt window to some other window by clicking in the window you want to use. The prompt window will stay open waiting for you to complete the operation. When you are ready to finish inserting the link, simply click in the prompt window to make it the active window, shown by the flashing crosshairs, and continue where you left off.

Insert Links

If you need to insert multiple links of the same link type at the same location in a source card, **Insert Links** saves you from inserting each link individually.

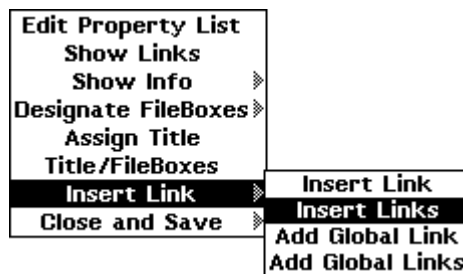


Figure 8-17. Card Menu the **Insert Links** Option on **Insert Link** Submenu Selected

Insert Links works exactly as **Insert Link** except you will be allowed to select multiple destination cards. It is possible to shift-select a mixture of text and link icons from a card. In this case, the source card's prompt window will ignore all the extraneous text. To complete the insert operation, either type a carriage return or select the **Done** menu option.

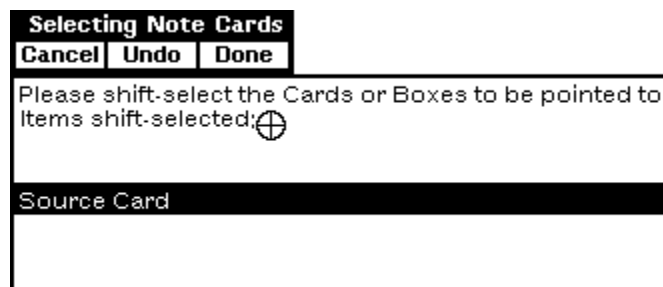


Figure 8-18. Selecting Note Cards Prompt Window

The **Cancel** option terminates the entire link insertion operation.

Undo allows you to remove the last item from the list. You can select **Undo** until you have removed all the items from the list.

Add Global Link

Global links are not rooted in a card's contents. For this reason we draw the contrast between *inserting* a link into a card's contents and *adding* a global link to a card.

A second result of this difference is that the procedure for **Add Global Link** is somewhat different from that for **Insert Link**. To add a global link, hold the left mouse

button down on the card's title bar and slide off **Insert Link** to access the submenu and select the **Add Global Link** option.



Figure 8-19. **Card Menu with Add Global Links Option on Insert Link Submenu Selected**

Because you are adding a link to a card itself and not inserting a link into a card's contents, it is not necessary to first position the caret cursor to select the link icon's insertion point. Note that the link will not be visible unless you use the **Show Links** option to show the links to and from that card. Aside from these few minor differences the procedure is exactly the same as that for **Insert Link**.

Add Global Links

Add Global Links functions just like a combination of **Add Global Link** and **Insert Links**. It allows you to add multiple global links of the same link type to a source card without adding each link individually. If you want the global links to have different link types, they must be added one by one using **Add Global Link**.

Creating Links in Other Card Types

There are three basic card types in NoteCards: text-based, sketch-based and graph-based cards.

Text-Based Cards

Text-based cards are based on the TEdit editor and can contain the output from sketch-based and graph-based cards. Text-based cards include Text, FileBox, Search, LinkIndex, and Document cards. The link creation procedure for all text-based cards is the same as for Text cards.

Sketch-Based Cards

Sketch-based cards are based on the Sketch editor. There is only one sketch-based card and it is the Sketch card.

The link icon insertion procedure differs from that for text-based cards only concerning *when* the new link icon is positioned in the card. For text-based cards, the cursor is positioned where you want the link to appear and then one of the link insertion options is selected. For sketch-based cards, the order is reversed. The menu option is selected first and then, after you have indicated which card you want to link to, you move the cursor into the Sketch card and position the link icon.

Graph-Based Cards

Graph-based cards are based on the graph editor. Graph and Browser cards are graph-based cards.

As with sketch-based cards, the link icon position is selected after the card you are linking to.

Deleting Links

When you delete a link icon you are deleting the to-link that points from the source card to the destination card and also the from-link which points from the destination card back to the source card.

To understand what happens when you delete a link, create a Source Card and a Destination Card. Next insert a link from the Source Card to the Destination Card, and open the **List of Links** windows on each card by selecting the **Show Links** option from the left-button title-bar menu. When you delete one link, all the links will be deleted.

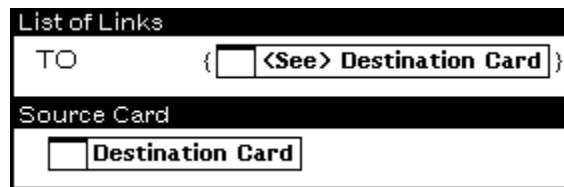


Figure 8-20. **List of Links** Window Showing To-link to Destination Card

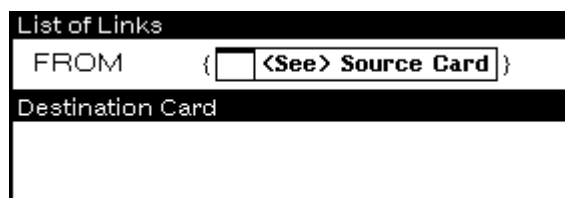


Figure 8-21. **List of Links** Window Showing From-link Pointing Back to Source Card

The delete procedure for links depends on where you are accessing the link and the card type you are deleting the link from.

Deleting Links from Card Contents

Text-Based Cards

Link icons in the body of text-based cards can be deleted in three different ways.

1. Select the link icon with the mouse by clicking in the *inactive* region of the link icon and pressing the delete key. Also, you can select a region of text in a text-base card with a sequence of left+right or middle+right mouse buttons and press the delete key to delete the selected region which can include a mixture of text and link icons.
2. Hold the control key down and select the link icon by pressing the left or middle mouse button in the link icon active region, when you release the control key, the link icon will be deleted.
3. Simply backspace over a link to delete it.

All of these operations can be undone by pressing the Undo key.

For more detail on inactive regions of link icons and coordinated keyboard-plus-mouse operations, see the Link Access and Use section above and as well as the TEdit documentation.

Sketch-Based Cards

You can delete link icons in the body of sketch-based cards by bringing up the sketch menu, if it is not permanently attached to the right side of the sketch window, and selecting the delete option. Next, click in the link icons control point which is the small box attached to one corner of the link icon.



Figure 8-22. Selected Link Icon in a Sketch Card

Press the UNDO key to undo this operation.

For more detail on this and other Sketch operations see the Sketch documentation.

Graph-Based Cards

To delete a link from a Graph card, hold the right mouse button down in the Graph card window, select the **Delete Node** option, and then click on the card you want to delete. When you confirm, the link will be deleted.

Note, this operation cannot be undone.

Because a Browser card is an overview of a link network in addition to being a member of that network, the meanings of delete operations in Browser cards is not straightforward. Please turn to the section on Browser cards in Chapter 11, System Cards, for a detailed explanation of links in Browser cards.

Deleting Links from the Show Links Display

Deleting links from the show links display is very similar to deleting links from text-based cards.

Select the link icon with the mouse by clicking in the inactive region of the link icon and pressing the delete or backspace key.

You can also hold the control key down and select the link icon with the mouse by pressing the left- or middle-mouse button in the link-icon active region, when you release the control key, the link icon will be deleted. This second way is frequently the easier of the two, as it does not require you to access the narrow inactive regions on either end of the link icon.

Note, this delete operation cannot be undone.

Tailoring Links

It is possible to modify the appearance of links to suit your personal tastes and the task at hand. Links can be modified using the middle-button icon menu and the NoteCards "System Parameters" menu.

Link Ops Menu

You bring up the "LinkOps" menu by holding the middle mouse button down in the active region of the link icon.

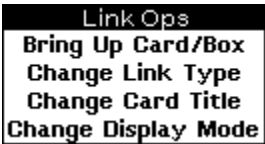


Figure 8-23. **Link Ops** Menu

Bring Up Card/Box	Performs the same function as clicking the left mouse button in the link icon's active region. It traverses the link to the destination card and opens it, or flashes it if it is already open.
Change Link Type	Allows you to choose a different link type for the link you select.
Change Card Title	Changes the title of the card the link points to and the titles shown in all the link icons pointing to that card. The one exception to this statement is when you have created one-way cross-file links. These links update only when you access the link icon which points to the changed card.
Change Display Mode	Selecting Change Display Mode brings up the Display Mode? Menu. This menu allows you to override the display mode defaults for the individual link icon you have selected.

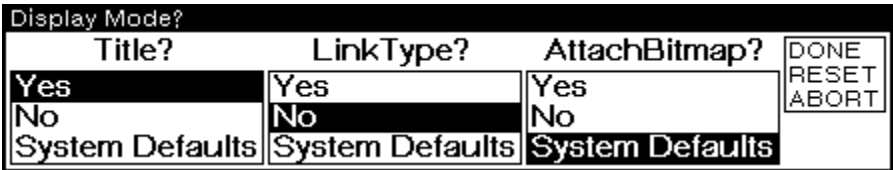


Figure 8-24. **Display Mode?** Menu

There are two levels of defaults for link icon display modes. Each card defined in the NoteCards system sets how links will be displayed in its contents. This is the first level of defaults and you have no control over them. These defaults can be set to three values, Yes, No and System Defaults. These are the values which appear in the **Display Mode?** menu shown above. If the value is set to System Defaults, NoteCards consults the System Parameters to determine whether or not to display that particular attribute. See Chapter 13 for how to set link default display modes.

With all the values set to **No**, the link appears as a small rectangular box.



Figure 8-25. Link Icon with All Display Mode Values Set to **No**

With all the values set to **yes**, the links have the following appearance.

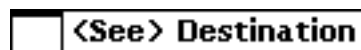


Figure 8-26. Link Icon with all Display Mode Values Set to **Yes**

The link type is shown in angled brackets and the card title is to the right of the link type. The card-type bitmap is the square piece on the left of the link icon and varies in design depending on the card type.



Represents Text cards.



Represents FileBox cards.
The image is a stylized file drawer.



Represents Sketch cards.
The image is a stylized sketch.



Represents Graph cards.
The image is a stylized letter "G."



Represents Browser cards.
The image is a stylized Browser.



Represents Search cards.
The image is a stylized eye.



Represents LinkIndex cards.
The image is a stylized list of link icons.



Represents Document cards.
The image is a stylized linear list of note cards.



Represents cards for which the type is unknown occurs with cross-file links.

System Parameters for Links

There are nine system parameters to enable you to customize the appearance of links. They are **Link Icon Font**, **Link Icon Border Width**, **Link Icon Multi Line Mode**, **Cross File Link Mode**, **Link Icon Max Width in Pixels**, **Link Icon Show Title**

Default, Link Icon Attach Bitmap Default, Link Icon Show Link Type Default, and Use Deleted Link Icon Indicators. For a complete discussion of these parameters, see Chapter 13.

Cross-File Links

The NoteCards system allows you to have links from one notefile that point to cards in another notefile. These types of links are known as cross-file links. See also Chapter 13, System Parameters for how to set the cross-file link mode.

Appearance

Cross-file links are indicated by an arrow bitmap on the right-hand side of the link icon.



Figure 8-27. Cross-file Link Indicated by Arrow Bitmap on Right Side of Link Icon

It is also possible to cross-file cards. This is to say that you can file a card from one notefile in the FileBox of another notefile. Note that this is only possible after you have first filed it in its origin notefile. This is, in effect, just another way of creating a cross-file link. Cards are always physically filed in their origin notefiles.

When a cross-file link is displayed, and the notefile the link points to is closed, then the link can have the following form.



Figure 8-28. Cross-file link to Closed Notefile (Question-Mark Bitmap indicates Card Cannot be Accessed to Determine Card Type)

The card-type bitmap on the left of the link icon is a question mark because the card type is stored on the card and this link has no access to the card because the notefile is closed. As a result, the link cannot display the appropriate card-type link. When you click on the cross-file link the system will ask you if you want to open the corresponding notefile. At this point, if you redisplay the card, by selecting the **Redisplay** option on the window menu, the link will be shown with its appropriate card-type bitmap.

Missing Notefiles

If you rename or delete a notefile which has cross-file links pointing to it, the system tries to open the notefile by its original name. When it cannot find the notefile under its original name, it asks you which notefile to look in. You cannot give a name with wild card characters like "*" or "?." You must give actual notefile names. The system continues to prompt you for new notefiles to search until you give up.

[This page intentionally left blank]

9. CARDS AND BANNERS

Cards are the basic units of information in the NoteCards system. This information can be a mixture of text and graphics. The cards can contain as much or as little information as you deem necessary. You can link each card to any number of other cards to create an information network which you can randomly browse or systematically read. You access cards through the notefile banner.

This chapter explains:

- What you can do from the notefile banner
- How to access old cards
- How to create new cards
- What card types there are
- What the card menu does

Notefile Banner

Each notefile which you have opened is represented on the screen by a notefile banner. The notefile name appears across the top of the banner in the region called the title bar (see Figure 9-1). The semicolon after the name is followed by the notefile version number.



Figure 9-1. Solar System Notefile Banner

The Banner has three active regions: the title bar, the **New Cards** option and the **Special Cards** option.

Banner Title Bar

From the title bar you can access three notefile menus.

Notefile Ops Menus

There are two Notefile Ops Menus. One displays only those options which work on open notefiles. The other displays only those options which work on closed notefiles.

Holding the left mouse button down in the Banner title bar when the notefile is *closed* brings up the following **Notefile Ops** Menu. The **New Cards** and **Special Cards** options are grayed over when the notefile is closed.

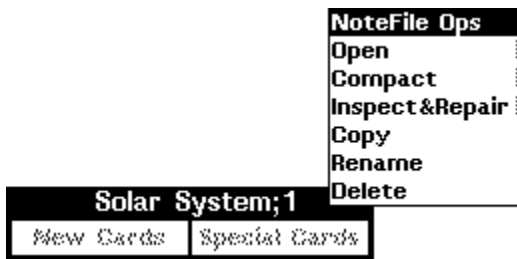


Figure 9-2. **Notefile Ops** Menu from Closed Notefile Banner

Holding the left mouse button down in the Banner title bar when the notefile is *open* brings up the following **Notefile Ops** Menu.

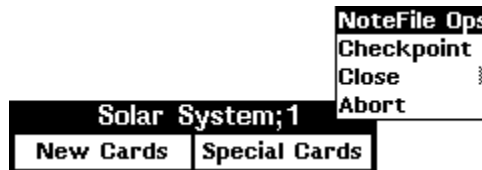


Figure 9-3. **Notefile Ops** Menu from Open Notefile Banner

These two sets of options are a subset of those found on the **Notefile Ops** menu accessed by holding the left mouse button down on the MenuBox Icon's **Notefile** option.

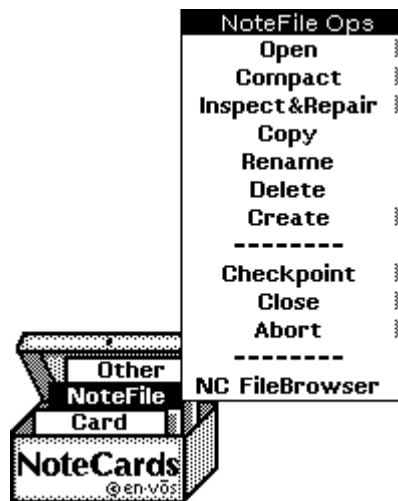


Figure 9-4. MenuBox Icon's **Notefile Ops** Menu

All of these commands are discussed in Chapter 12.

Middle-Mouse-Button Title-Bar Menu

Holding the middle mouse button down in a notefile Banner's title bar brings up the following menu.

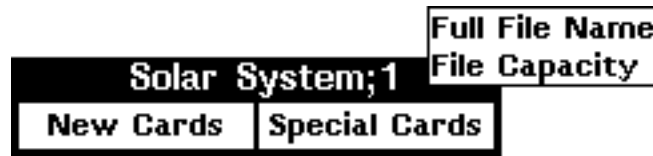


Figure 9-5. **Middle-Button** Menu in Notefile Banner Title Bar

Full File Name

Selecting the **Full File Name** option will bring up a window containing the full name of a file, including its directory path.



Figure 9-6. Full File Name of Solar System Notefile

Press any mouse button to continue. The window closes and you are allowed to continue your work. This option is useful when you have two files with the same name, but different directories, on the screen and you need to be able to differentiate their banners.

File Capacity

When the notefile is open, selecting the **File Capacity** option attaches a window to the top of the notefile Banner which displays the percentage and ratio of used cards in the notefile.

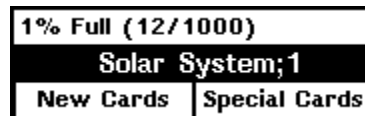


Figure 9-7. Notefile Banner with File Capacity Window Open

To close the file capacity window, hold the right button down in the file capacity window, not in the Banner title bar, and select the **Close** option off the window menu.

When 90% of the cards in a notefile have been used, the file capacity window will automatically open and warn you that the file is almost full and that it needs to be compacted. You must close a notefile before you can compact it. See Chapter 12, MenuBox Icon, for instructions on compacting notefiles.

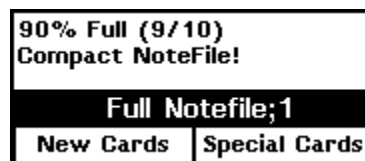


Figure 9-8. Notefile Banner with File Capacity Window Warning to Compact Notefile

If you try to create a new card when the notefile is 100% full, the system will display a message that the notefile is full and you will be asked if it is ok to checkpoint the notefile and make room for some number of new cards.

Expand notefile index?		
Yes	Cancel	Change Num
{DSK}/usr/users/kmount/NC/NF/Full Notefile.NOTEFILE Okay to checkpoint notefile and make room for 5 new cards?		
100% Full (10/10) Compact NoteFile!		
Full Notefile;1		
New Cards	Special Cards	

Figure 9-9. Expand Notefile Index? Menu and Prompt Window

If you select **Yes**, the system saves all the open cards, closes the notefile, expands the notefile index by the amount you indicated, reopens the notefile, and creates the card you requested. The notefile is not compacted in this operation.

Cancel stops the operation. **Change Num** allows you to change the number of cards to expand the notefile by, after which you can choose the **Yes** option from the **Expand notefile index?** menu and proceed to save and expand the notefile.

Special Cards

Old notecards are accessed from the notefile Banner under the option **Special Cards** (see Figure 9-1). Clicking the left mouse button in the **Special Cards** option brings up the "Table of Contents" FileBox card for the named notefile.

Solar System;1 {DSK}/usr/users/kmo				
Table of Contents				
FILE BOXES				
<table border="1"><tr><td>☐</td><td>Sun</td></tr><tr><td>☐</td><td>Planets</td></tr></table>	☐	Sun	☐	Planets
☐	Sun			
☐	Planets			
NOTE CARDS				

Figure 9-10. Table of Contents FileBox Card for Solar System Notefile

Holding down the middle mouse button in the **Special Cards** option brings up the **Special Cards** menu.

Special Cards
Table of Contents
Orphans
To Be Filed

Figure 9-11. Special Cards Menu

This menu allows you to access all of the top-level FileBoxes. Ultimately, all accessible cards are linked to one of these three FileBoxes.

Table of Contents FileBox

This is the top-level FileBox of each notefile. It is intended for storage of links to FileBoxes and other cards at the highest level of the information hierarchy.

To Be Filed FileBox

A temporary FileBox for cards that are not filed in any FileBoxes. If you close a card without specifying its FileBox, or if you close a notefile, without having specified the FileBoxes for every card, the system places the cards without assigned FileBoxes in the "To Be Filed" FileBox.

If you later designate a FileBox for an unfiled card using the **Designate FileBoxes** or **Title/FileBoxes** commands from the card menu, this does not remove the card from the "To Be Filed" FileBox. To remove a card from the "To Be Filed" FileBox either delete its link or use the **Unfile From FileBoxes** suboption on the card menu of the newly filed card.



Figure 9-12. Card Menu Showing **Unfile from FileBoxes** Suboption Selected

As a general rule, it is a good idea to check the "To Be Filed" FileBox at the end of every session to verify that you have properly filed every card where you want it. You might also keep a copy of this FileBox on the screen so that you can tell at a glance if any cards have been filed there.

Orphans FileBox

A FileBox for cards whose last link from another FileBox has been removed.

Removing a link icon from the "Orphans" FileBox severs the card from the FileBox hierarchy but does not delete it from the notefile. Cards in this state are lost cards. The only means of retrieving a lost card is with a search card.

As a general rule, it is a good idea to check the "Orphans" FileBox at the end of every session to verify that you have properly filed every card where you want it. You might also keep a copy of this FileBox on the screen so that you can tell at a glance if any cards have been filed there.

New Cards

New cards are created from the notefile Banner under the option **NewCards** (Figure 9-1).

Clicking the left mouse button on the **NewCards** option immediately creates a specific type of card. The type of card created is determined by the value of the **Default Card Type** System Parameter. See Chapter 12 for a description of how to set this parameter. The default value is **Text**.

Pressing the middle mouse button over the **NewCards** option brings up the **Card Types** menu.

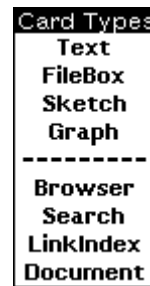


Figure 9-13. **Card Types** Menu

Select one of these options to create a card of that type.

Figure 9-14 graphically represents the notecard types and their range of use. Text, Graph, and Sketch cards help you to collect and express your ideas. FileBoxes allow you to express the relationships between your ideas and organize them into coherent structures. Browser cards provide a broad range of functionality, and are a means of representing as well as retrieving information. LinkIndex cards build sorted lists of cards. Search cards perform searches on card titles. Document cards create a linear document from your linked cards.

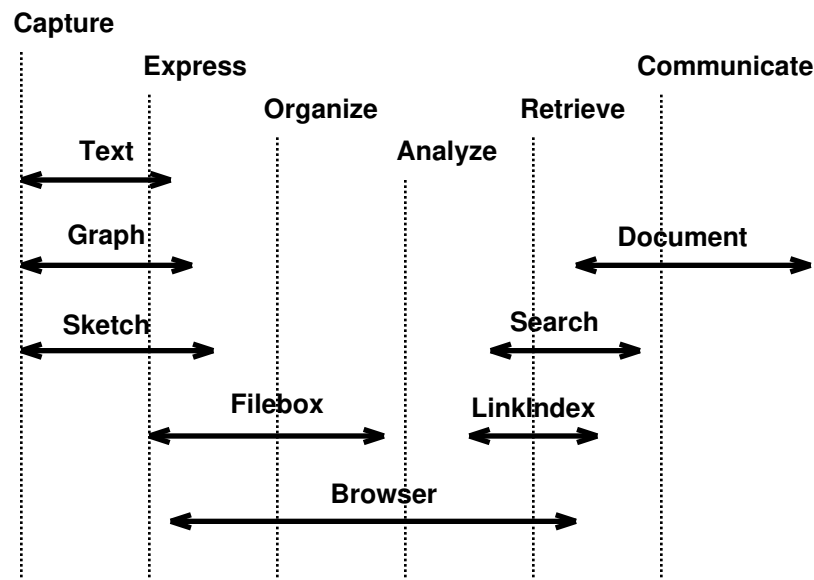


Figure 9-14. Card Types and Their Range of Use

User Cards and System Cards

Cards can be broadly divided into two categories. User cards are those cards for which you create the contents. System cards are cards where the contents are built by the system for you. Chapter 10 discusses user cards; Chapter 11, system cards.

In Figure 9-13, the top four card types (Text, FileBox, Sketch, and Graph) are user notecards. The bottom four card types (Browser, Search, LinkIndex, and Document) are system notecards.

Text-, Sketch-, and Graph-Based Cards

All cards are based on one of three editors, the text editor TEdit, the Sketch editor Sketch, or the Graph editor Grapher.

Text, FileBox, Search, LinkIndex, and Document cards are all based on the text editor TEdit. Hence they all behave in essentially the same way and are referred to as text-based cards.

The Sketch card is the only card based on the Sketch editor. This card is referred to as a sketch-based card.

Graph and Browser cards are both based on the graph editor Grapher. These cards behave in basically the same way and are referred to as graph-based cards.

Card Menu

All cards have a left-button title-bar menu called the card menu. This menu is the same for all cards except FileBox cards. Each card also has a middle-button title-bar menu which is frequently card-specific. These menus are discussed under their respective card types in Chapter 10, User Cards, and Chapter 11, System Cards.

Standard Card Menu



Figure 9-15. Note card Menu

This menu is found on all cards except the FileBox card which has a slightly different version.

Edit Property List

The property list editor allows you to associate property-value pairs with cards. For example, you might want to attach a Certainty Value property (and value) to every card. The value of this property would indicate the degree to which you believe the information contained on the card is true.

You bring up the **Edit Property List** window by selecting the **Edit property List** option off the card menu.



Figure 9-16. Edit Property List Window

You can only change the property list of a card interactively, from the menu shown in Figure 9-17. Holding down the left or middle mouse button in the **Edit Property List** window's title bar brings up the following edit menu for property lists. Select commands from this menu before releasing the button.



Figure 9-17. Menu from Edit Property List Window Title Bar

- | | |
|---------------------------------|--|
| Add New Property | Adds user-defined properties to the property list of this card. Type in the property name and the value when prompted. Next, select one of the properties already in the list. The new property will be inserted in front of the selected property. Properties are displayed in bold type and values are displayed between brackets in regular type. You can abort the add process by hitting the Stop key.

In Figure 9-16, Source and Certainty Factor are the properties and "John Smith" and ".9" are the values. You can edit the values directly with the mouse and keyboard as you would any text string. |
| Delete Selected Property | Deletes a property from the property list of this card. After choosing this option click on the property to be deleted with the left mouse button. Properties are displayed in bold type. Once you have started the delete process there is no way to abort it, but you can hit the Undo key to undo the deletion. |
| Quit w/o Saving Changes | Closes the display without saving any of the current changes made using Add New Property or Delete Selected Property . |
| Quit - Saving Changes | Closes the display saving all current changes made using Add New Property or Delete Selected Property . |

Show Links

Displays, in the **List of Links** window above the card, a list of all links to and from other cards. Links are represented by link icons. Selecting an icon in this window with the left mouse button displays the card referenced by that link icon.

To close this display, place the cursor in the title bar of the **List of Links** window, press the left button, and select **Quit** from the single item menu.

For more on the **Show Links** option, see the **Viewing Local and Global Links** section in Chapter 8.

Show Info

Brings up the **Card Attributes** window. This window displays the card's type, the dates each of the card parts was last changed, and a list of dates chronicling when the card was updated.

Card attributes	
Type	Text
ItemDate	" 5-Jan-89 18:00:19"
TitleDate	" 4-Jan-89 09:55:45"
LinksDate	" 6-Jan-89 18:04:26"
PropsDate	" 5-Jan-89 18:00:20"
Updates	((MOUNTFORD,ENVOS " 5-

Figure 9-18. **Card Attributes** Window

The **Show Info** option also automatically brings up the notefile indicator on the card.

Indicate NoteFile This pull-across subitem on the **Show Info** option brings up the notefile indicator without bringing up the **Card Attributes** window.

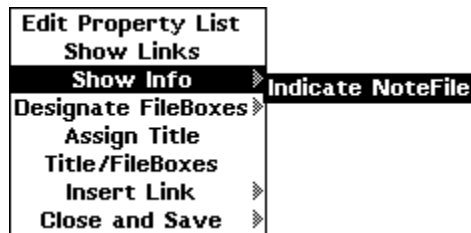
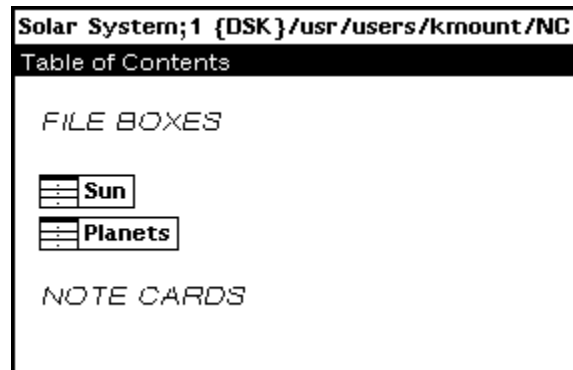
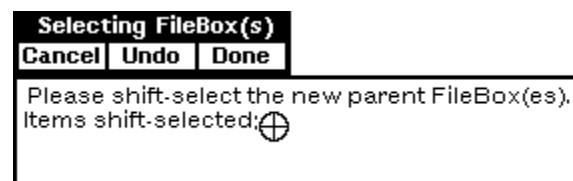
Figure 9-19. **Indicate Notefile** Submenu of **Show Info**

Figure 9-20 Card with Notefile Indicator Window Open on Top of Card

To close the **Card Attributes** window, just choose the **Close** item on the standard window title bar menu of the **Card Attributes** window. Closing the **Card Attributes** window does not close the notefile indicator window, which must be closed separately using its standard window title bar menu.

Designate FileBoxes

Pops up a prompt window above the card asking the user to file the card in one or more FileBox cards.

Figure 9-21. **Selecting FileBox(es)** Prompt Window

Associated with the prompt window is a three-item menu. Select **Done** from this menu or type a carriage return after shift-selecting the FileBoxes to file the card in. Select **Cancel** to abort the filing operation. Choose **Undo** to remove the last selected FileBox from the list of new parent FileBoxes.

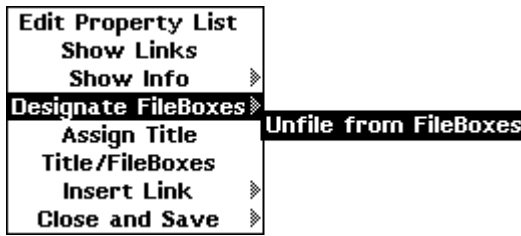


Figure 9-22. Card Menu with **Unfile from FileBoxes** Suboption Selected

Unfile from FileBoxes

Unfiles the card from the FileBoxes you select. You unfile the card by shift-selecting the FileBox card IDs from their title bars. The system tells you when you have made an invalid selection. The prompt window menu options are the same as those for **Designate FileBoxes**, shown in Figure 9-21.

Assign Title

Allows you to assign a title to a card or edit an existing title.

You can reposition the cursor in the edit string with the mouse. You can also use the mouse to delete pieces of the string by using the left mouse button to position the cursor at the beginning of the piece you want to delete and sweeping out the portion to delete while holding down the right mouse button. When you release the right mouse button the selected text will be deleted.

To cancel the delete operation, still holding the right mouse button down, move the mouse cursor outside the prompt window and release the right mouse button.

You can undo the delete operation by hitting the Undo key.

Title/FileBoxes

Selecting **Title/FileBoxes** is identical to selecting **Assign Title** and **Designate FileBoxes** in sequence.

Insert Link

Inserts a user-specified link to another card inside the body of the current card. The link is represented by a link icon.

Before execution of this command, select the point, in the body of the card, where you want the link icon to appear.

When you select this menu option, a menu pops up displaying a list of link types currently available in the notefile. Link types are notefile specific. Specify the type of link by selecting one from this list or select **--New Link Type--** to create a new type. This new type of link is added to the notefile and becomes a choice in its list of link types. It is not possible to assign a system-reserved type of link to a user-specified link. Select ****CANCEL**** from the menu to abort the **Insert Link** command or just click outside the menu.

After a type has been designated, a prompt window and menu pops up above the card asking the user to choose the destination card for the link. The user has the option of selecting an existing card or of creating a new card as the destination card by selecting **New Card** from the menu with the left mouse button. If this option is chosen, a menu of card types pops up from which the user selects the desired type of card. Again, selecting **Cancel** from the "Selecting Note Card" menu aborts this command.

At this point, the link icon will be inserted at the flashing caret in the body of the source card.

For more on **Insert Link** and its submenu options **Insert Links**, **Add Global Link**, and **Add Global Links**, see Chapter 8.

Close and Save

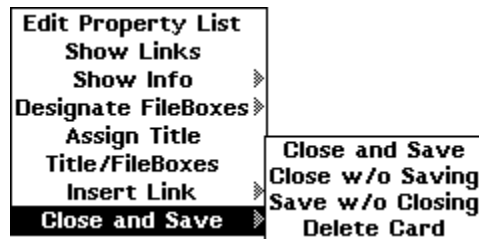


Figure 9-23. **Close and Save** Submenu

Close and Save

Saves the card in the notefile before closing the card. If the **Force Titles** and **Force Filing** system parameters are set to **Yes**, NoteCards will request that you provide a title and a FileBox for the card before closing it. If you do not wish to provide this information, simply type a carriage return to both requests. In this case, the card will be titled "Untitled," and be filed in the "To Be Filed" FileBox.

Close w/o Saving

Closes the card without saving it, in the notefile. Any changes you have made to the card since the last save are lost. This option is useful if the card contents are mistakenly lost or scrambled while you are editing. This command may ask for confirmation. Type a carriage return or click on the **Yes** option to confirm. To cancel, type **n** or **N** and RETURN, or click on the **No** option.

Save w/o Closing

Saves, to the notefile, all changes you have made to the card without closing the card. Saving updates the card in the notefile but does not update the notefile index. If there is a system crash you will have to perform an **Inspect & Repair** operation to recover the card contents.

Delete Card

Permanently deletes the card from the notefile and all its links to and from other cards. Because this deletion is irreversible, the user is asked to confirm before the delete command is executed. Type a carriage return or click on the **Yes** option to confirm. Type **n** or **N** and RETURN, or click on the **No** option to cancel. Note that a deleted card cannot be retrieved with a search card as a lost card can.

FileBox Card Menu

The FileBox card menu differs in only two items, **Add Global Link**, and **Put Cards Here**.

Add Global Link

The **Add Global Link** and **Add Global Links** options are the same global link options found on the submenu of the **Insert Link** option. **Insert Link** appears in this position on this menu for all other cards. These two options are discussed extensively in Chapter 8.



Figure 9-24. FileBox Card Menu with Add Global Link Submenu

Put Cards Here

The **Put Cards Here** menu option is a specialization of the **Insert Links** option for FileBox cards. It allows you to file more than one card in a FileBox in a single operation. Selecting this option brings up the **Selecting cards to file** prompt window into which you can shift select the card IDs from the title bars of all the cards you want to file in this FileBox card.

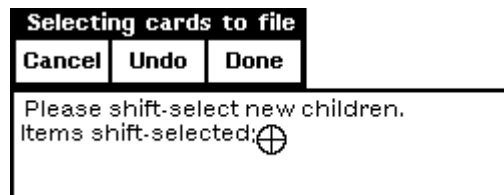


Figure 9-25. Selecting Cards to File Prompt Window

[This page intentionally left blank]

10. USER CARDS

Cards can be broadly divided into two categories. User cards are those cards for which you create the contents. System cards are cards where the contents are built by the system for you. This chapter discusses user cards.

This chapter explains how to use:

- Text cards
- FileBox cards
- Sketch cards
- Graph cards and the graph editor
- Bitmap editor

Text Cards

The Text card is based on TEdit, a versatile editor and text formatter. This entire guide was produced using TEdit. For a detailed discussion of how this editor works, see *A User's Guide to TEdit*. Text cards allow you to include sketches, graphs, and bitmaps. To learn how to manipulate sketches, see *A User's Guide to Sketch*. For graphs and bitmaps see the sections below on the Graph card and the bitmap editor.

Text-Card Menu

The text-card menu is the same as the TEdit menu with one additional option (Restart Editor) separated from the others by a dashed line (see Figure 10-1).

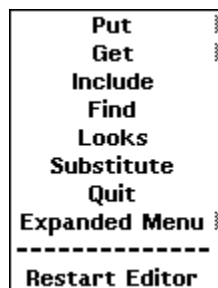


Figure 10-1. **Text-card** Menu

Use the **Restart Editor** command when the contents of the Text card are incorrectly displayed on the screen.

For all the other menu items, see *A User's Guide to TEdit*.

FileBox Cards

The FileBox card, like the Text card, is based on TEdit. For a detailed discussion of how this editor works, see *A User's Guide to TEdit*. Text-based cards allow you to include sketches, graphs, and bitmaps. To learn how to manipulate sketches, see *A User's Guide to Sketch*. For graphs and bit maps see the sections below on the Graph card and the bitmap editor.

A FileBox is a card that contains links to other cards including other FileBox cards. All cards, including FileBox cards, can be filed in one or more FileBoxes. Every card, including FileBox cards but excluding the top level Special FileBox Cards, is contained in at least one other FileBox. Whereas other cards may be linked together to form an arbitrary network, the set of FileBoxes forms a strict hierarchy. This is to say that no child FileBox is allowed to have its parent FileBox as a child. In short, no circular linkages can exist.

FileBoxes hold all cards relating to a given topic. A FileBox typically contains both links to subFileBoxes, which contain any cards relevant to the subtopics of the main topic, and links to other card types, which contain information relevant to the main topic. For example, the screen image in Figure 10-2 shows a FileBox containing both FileBoxes (Sun and Planets) and other note cards (Overview), all dealing with the main topic of Solar System.

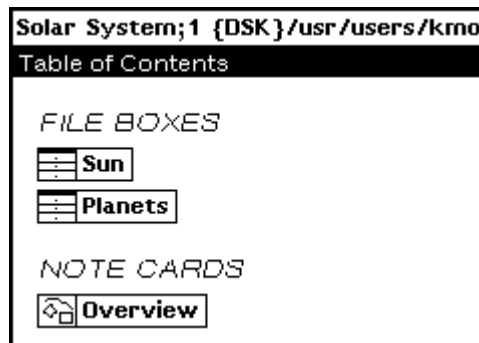


Figure 10-2. Solar System FileBox containing SubFileBoxes and Sketch Card

The FileBox structure provides a way of keeping track of sets of cards on a common topic. In contrast, the links between individual cards allow you to represent the interconnections between various ideas or pieces of information, independent of any categorization into topic areas.

The markers FILE BOXES and NOTE CARDS help differentiate the kinds of cards filed in the FileBox. In addition, since FileBoxes are text-based cards, anything you can do with a text card, you can also do with a FileBox. For example, you can insert your own labels or short lines of commentary to break up the links into subgroups.

Suggested FileBoxes and Note Cards

You may find it helpful to create the following types of general FileBoxes.

- Bibliography – a FileBox for the collection of sources used in the notefile.
- Index – a FileBox listing keywords from the notefile; it may be helpful when using Search cards.

- ReadMe – a note card in the top level FileBox giving global information about the notefile for first-time browsers.
- Active Cards – a FileBox kept at the top level of the FileBox hierarchy containing FileBoxes and note cards that represent work in progress and are thus frequently accessed. A Sketch card containing links to these FileBoxes and note cards is another method of organizing active cards, using spatial cues as a way of representing structure.

FileBox-Card Menu

This menu is the same as the Text-card menu. See the section immediately above on Text cards as well as *A User's Guide to TEdit*.

System Parameters Affecting FileBoxes

FileBoxes have two system parameters associated with them, **Markers In FileBoxes** and **Alphabetized FileBox Children**. For a complete discussion of these parameters see Chapter 13.

Sketch Cards

The Sketch card is based on Sketch, a sophisticated graphics package. For a detailed discussion of how Sketch works, see *A User's Guide to Sketch*. Sketch cards allow you to include graphs, and bitmaps. To learn how to manipulate graphs and bitmaps see the sections below on the Graph card and the bitmap editor.

Sketch-Card Menu

This menu (Figure 10-3) is the same as the Sketch editor menu. For a detailed discussion of this menu's functionality, see *A User's Guide to Sketch*.



Figure 10-3. Sketch Card Menu

System Parameters Affecting Sketch Cards

Sketch cards have one system parameter associated with them, **Attach Sketch Menu**. For a complete discussion of this parameter see Chapter 13.

Graph Cards

The Graph card allows you to construct a layout of user-defined words or phrases, called nodes, which may be connected together with lines to indicate some structure. Each node may be easily moved about the card without losing its connections.

Graph-Card Menu

The Graph card, like the Browser card, is based on the graph editor Grapher. To make the Graph-card menu and terminology more consistent with the Browser-card menu and terminology, the Graph card presents a slightly different menu to you than Grapher does. However, since the functionalities are virtually identical, both the Graph-card and Grapher menus are discussed in parallel below. The graph-card menu-option titles are left justified with corresponding Grapher menu-option titles in parentheses.

Display the **Graph-card** menu by pressing the right mouse button in the body of the card. Select the desired command before releasing the button.

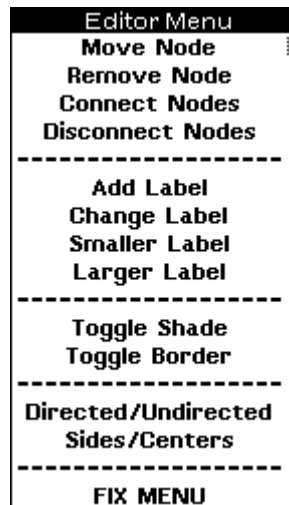


Figure 10-4. **Graph-card** Menu

Grapher Menu

You will probably encounter the Grapher menu only if you save your document cards to TEdit files and edit a graph-card graph from within the TEdit document. You can safely skip over this section and still understand the Graph card.

To edit a graph from a TEdit document, hold any mouse button down in the graph region. Then select the **Edit graph** option from the one-item menu (see Figure 10-5).

Edit graph

Figure 10-5. **Edit Graph** Menu

When you select this option, Grapher opens a window containing the graph. Hold the middle mouse button down in this window to make the menu shown in Figure 10-6 appear. Use the left mouse button to move nodes. When you are done editing the graph, select the **STOP** option.

While you are editing a graph, Grapher captures the type-in process and does not allow you to do anything other than mouse operations. To free the type-in process, select the **STOP** option.

There is no simple way to abort out of the Grapher editor and throw away all the changes you have made to the graph. For this reason, if you are going to edit the graph extensively, we recommend that you use shift-select to copy the original graph in place and call the Grapher editor on the copy. In a worst case scenario, try pressing the STOP key or typing CONTROL-E.

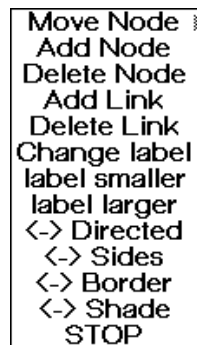


Figure 10-6. **Grapher** Menu

Prompts for information or confirmation are given in the card prompt window for Graph cards and appear in the system prompt window for Grapher. General information is printed to the system prompt window for both Graph cards and Grapher.

Graph-Card and Grapher Menu Options

The menu option at the left side of each paragraph below refers to the Graph-Card Menu. The option in parentheses refers to the Grapher Menu.

Move Node (Move Node)

Moves a node and connections to a new position. After selecting this option, point to the node you want to move, press and hold the left mouse button, move the node to its new position, and release the mouse button.

Move Node has three options on a submenu (see Figure 10-7). These same three options appear on the Grapher **Move Node** submenu.

Figure 10-7. **Move Node** Submenu

Move Single Node	Functions exactly as Move Node does.
Move Node & SubTree	Moves a selected node and all subnodes to which it is connected. This operation does not move any super nodes of the selected node. That is, nodes which are connected to the selected node as opposed to nodes to which the selected node is connected.



Figure 10-8. Node and Subtree Selected to be Moved

Figure 10-8 shows what happens when you select **Move Node & SubTree** and then hold the left mouse button down on node B. Nodes B, C, and D are selected to be moved, but not node A. Grapher keeps track of the nodes where each connection starts and ends, and in this case the connections run from A to B to C to D. So node A is not in the subtree of node B and hence is not moved. See the **Directed/Undirected** option below for more information.

Move Region	Allows you to sketch out a region of the graph which you want to move. Move Region does not pay attention to the graph hierarchy, it only pays attention to the area you sweep out in the Graph card.
--------------------	--

Remove Node (Delete Node)

Removes a node from the graph. Select the node to be deleted with the left mouse button. The card prompt window prompts you for confirmation.

Connect Nodes (Add Link)

Draws a connection between two nodes. Select the **from** node and then the **to** node with the left mouse button when prompted. If a second overlapping connection is made running in the opposite direction between the same two nodes, the lines representing the connections, between those two nodes, will not be visible. You can make them visible by choosing the directed display option.

Figure 10-9. **Undirected** Option Graph (Connections from A to B to C and a third connection from C to B)

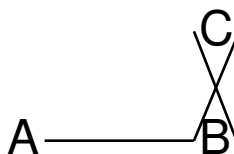


Figure 10-10. **Directed** Option Graph (connections from A to B to C and a third connection from C to B)

Disconnect Nodes (Delete Link)

Removes a connection from between two nodes. Select the **from** node and then the **to** node with the left mouse button when prompted.

Add Label (Add Node)

Pops up a window prompting you to type in a label name followed by a carriage return. The label appears next to the cursor within the graph card. Position the new label by moving the cursor to where you want the label to appear. Plant the label by clicking any mouse button. Cancel this operation by typing a carriage return before typing any other characters to the prompt window.

Change Label (Change Label)

Allows you to change a label. **Change Label** first waits for you to select, with the left mouse button, the label you want to change. It then pops open a window prompting you to type in a new label name followed by a carriage return. The new label immediately replaces the old label, preserving font, position, and connections. You can cancel this operation by clicking outside a node, or by typing a carriage return before typing any other characters to the prompt window.

Smaller Label (label smaller)

Decreases the font size of the selected node. Repeat this command as many times as necessary to achieve the font size you want.

Larger Label (label larger)

Increases the font size of the selected node. Repeat this command as many times as necessary to achieve the font size you want.

Toggle Shade (<-> Shade)

Inverts the shade around the selected node. For example, a black label on a white background becomes a white label on a black rectangular background. Select the node to be inverted with the left mouse button. To change the shade back, re-apply this option.

Toggle Border (<-> Border)

Draws a rectangular border around the selected node. Select the node to have a border drawn around it with the left mouse button. To remove a border, re-apply this option.

Directed/Undirected (<.> Directed)

A graph is stored as a directed lattice. Connections always run from one node to some other node.

The **Directed** option makes the flow of the connections explicit in the presentation of the graph. When you select the **Directed** option, connections prefer to run from the left side of the parent node to the right side of the child node when you have **Sides** selected (see Figure 10-11). When you have the **Centers** option selected, connections prefer to run from the bottom center of the parent node to the top center of the child node (see Figure 10-12).

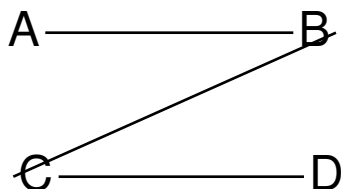


Figure 10-11. **Directed** Graph (shows flow of connections from A to B to C to D, with the **Sides** option selected)

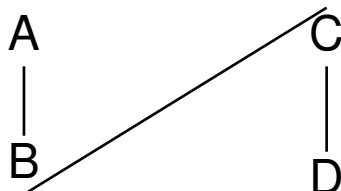


Figure 10-12. **Directed** Graph (shows flow of connections from A to B to C to D, with the **Centers** option selected)

The **Undirected** option draws the graph without regard to the flow of connections. The lines are drawn starting and ending on the sides of the nodes closest to each other (see Figures 10-13 and 10-14).

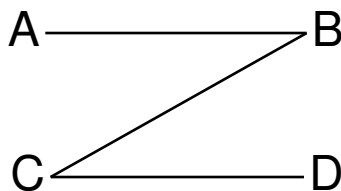


Figure 10-13. **Undirected** Version of Figure 10-11

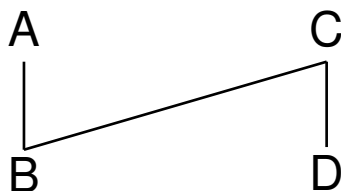


Figure 10-14. **Undirected** Version of Figure 10-12

Sides/Centers (<-> Sides)

The sides mode predisposes the graph editor to make the left and right sides of the nodes the connection points for lines, as shown in Figure 10-15.

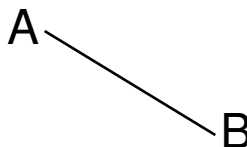


Figure 10-15. Graph Favoring Sides

The centers mode predisposes the graph editor to make the top and bottom centers of the nodes the connection points for lines, as shown in Figure 10-16.

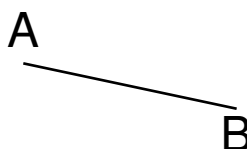


Figure 10-16. Graph Favoring Centers

FIX MENU

Attaches the graph-card menu to the right side of the Graph card.

(STOP)

Only appears on the Grapher menu, not on the Graph card menu. **STOP** exits the Grapher editor and saves all your changes.

Bitmap Editor

The bitmap editor allows you to manipulate bitmaps that have been inserted in Text or Sketch cards as well as TEdit and Sketch files. It is automatically invoked when the bitmap area is selected.

Inserting Bitmaps into Cards

The method for inserting bitmaps into text-based cards differs slightly from that for inserting them into sketch-based cards. Each procedure is discussed below.

Note: You cannot insert bitmaps into graph-based cards.

Text-Based Cards

Inserting a bitmap into a text-based card involves several steps.

1. Position the caret cursor where you want the bitmap to appear in the destination card, or TEdit window, by clicking at that position with the left mouse button.

2. Press the COPY key or either of the SHIFT keys, hold the right mouse button down somewhere in the background and select the **Snap** option from the single item menu which will appear (see Figure 10-17).



Figure 10-17. **Single-Item Snap Menu**

3. At this point the mouse cursor changes to look like this, . This is the prompt asking you to sweep out an area of the screen to be made into a bitmap.
4. Press and hold the left mouse button while you sweep out a region of the screen.
5. When you release the left mouse button, the bitmap is transferred to the designated card or edit window.

If you need to adjust the area you are sweeping out, hold down both the right and left mouse buttons to bring up the forceps prompt, . This prompt allows you to change corners so that you can adjust the size of the bitmap in all directions.

Sketch-Based Cards

The procedure for inserting bitmaps into sketch-based cards differs in only one respect from that for text-based cards. The sketch-based card must be the active card, which is to say you must click in it so that it has the type-in process to mark it as the destination for the bitmap. However, when you do this, you are not indicating the insertion point for the bitmap. In sketch-based cards, positioning the bitmap is done last. After you have swept out a region of the screen to include as a bitmap in the sketch, move the mouse cursor back into the Sketch card. When you enter the Sketch card, the snapped bitmap appears attached to your mouse cursor, and you can position it by clicking the left mouse button.

Bitmap Operations

Bring up the Operations on Bitmaps menu (shown in Figure 10-18) by moving the mouse cursor into a bitmap and holding down the left or middle mouse button.



Figure 10-18. **Operations on Bitmaps Menu**

Change Scale

Changes the scale or size of the bitmap. A scale of 2 doubles the size of the bitmap; a scale of .5 halves the size of the bitmap. To achieve the best results when shrinking or enlarging a bitmap, change the scale by evenly divisible amounts; for example, 4, 2, 1, .5, or .25.

Hand Edit

Invokes the bitmap editor on the bitmap. The bitmap editor is described in detail below.

Trim

Trims the white columns and rows from all four edges of the bitmap. This is a very useful operation to remove any extraneous white space from around the bitmap. Position the image that you are taking a snap of on a white background to take the greatest advantage of this option.

Reflect Left-to-right

Flips the bitmap about its vertical centerline.

Reflect Top-to-bottom

Flips the bitmap about its horizontal centerline.

Reflect Diagonally

Flips the bitmap about its X=Y diagonal. The resulting bitmap is reversed and lying on its right side. The same effect can be achieved by performing a **Reflect Left-to-Right** followed by a **Rotate Right**.

Rotate Left

Rotates the bitmap by 90 degrees in a counterclockwise direction. The resulting bitmap is lying on its left side.

Rotate Right

Rotates the bitmap by 90 degrees in a clockwise direction. The resulting bitmap is lying on its right side.

Expand on Right

Adds white space to the right of the bitmap. Specify the width of the white space in pixels using the number pad. Select **ok** when you are done.

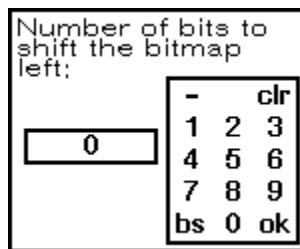


Figure 10-19. Number Pad

The number pad is used much like a simple calculator to enter numbers. The **ok** button returns the number to the system. **bs** deletes the last digit you entered. **clr** resets the input to 0. Abort the operation by setting the input value to 0 and selecting **ok**.

To enter a negative number, enter the digits and then select the minus sign. Entering a negative number removes the specified number of pixels from the right side of the bitmap.

Expand on Left

Adds white space to the left of the bitmap. Specify the width of the white space in pixels using the number pad. Then select **ok**. See the **Expand on Right** option for more detail.

Expand on Bottom

Adds white space to the bottom of the bitmap. Specify the width of the white space in pixels using the number pad. Then select **ok**. See the **Expand on Right** option for more detail.

Expand on Top

Adds white space to the top of the bitmap. Specify the width of the white space in pixels using the number pad. Then select **ok**. See the **Expand on Right** option for more detail.

Switch Black & White

Inverts all of the pixels in the bitmap; exchanges black for white and white for black.

Add Border

Adds a border to the bitmap. The system prompts for the width of the border using the number pad described above. It then prompts for the texture of the border with the texture bitmap editor.

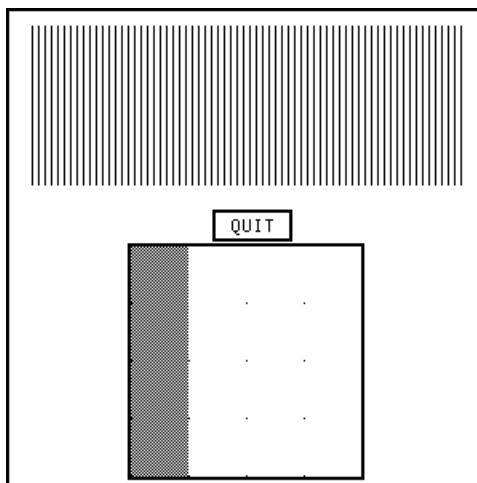


Figure 10-20 Texture Bitmap Editor

The area at the top of the window shows what the texture looks like in true screen scale and the bottom area contains a four-by-four edit array (see Figure 19-20). Click the left mouse button in the edit grid to turn a pixel on; click the middle button to turn a pixel off. Select the **Quit** option when the texture looks the way you want it to. The texture then appears as the border around the bitmap.

To abort this operation, select **clr** followed by **ok** on the number pad.

There is no simple way to abort this operation once you have brought up the texture bitmap editor. However, you can turn all the pixels off (set them to white space) and select **Quit**. Then select **Trim** from the **Operations on Bitmaps** menu to return the bitmap to its previous condition.

Bitmap Editor

The editing window has three active areas: a grid edit area in the lower part of the window, a display area in the upper left part, and a gray bar in the upper right (see Figure 10-21).

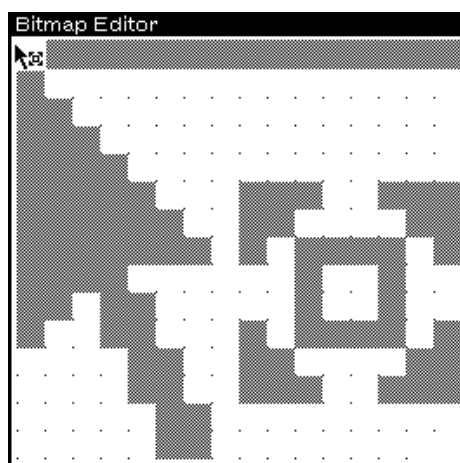


Figure 10-21. Bitmap Editor Window

In the edit area, the left button adds points and the middle button erases points. The display area shows the actual size and form of the bitmap. The gray bar provides access to the Bitmap Editor menu.

The right mouse button brings up the normal window menu in all areas of the window.

If the bitmap is too large to fit in the edit area, change the portion which can be edited by scrolling up and down in the left margin, and left and right in the bottom margin. Press the middle mouse button while in the display area to bring up a menu that allows you to make a global placement of the portion of the bitmap which can be edited. To see more of the bitmap you are editing, reshape the window to make it larger, or use the **GridSize** command, described below, to reduce the bit size in the edit area.

Whenever you press the left or middle mouse button down with the cursor inside the display area or the gray bar, the section of the bitmap that is currently in the edit area is shown in reverse video. Press the left button while in the gray bar to put the lower left 16 x 16-bit section of the bitmap into the mouse cursor for as long as the left button is held down.

Press the middle button while in the grey bar or in the title bar to bring up the **Bitmap Editor** menu (see Figure 10-22).

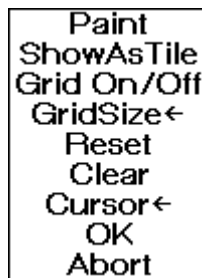


Figure 10-22. **Bitmap Editor** Menu

Hold the middle button down over a command to display an explanatory message in the system prompt window.

Paint

Puts the current bitmap into a window and calls the paint command on the bitmap. Use the left mouse button for drawing and the right for erasing. The paint command implements drawing with various brush sizes and shapes but only on an actual sized bitmap. To set brush characteristics and exit paint, press the right mouse button and select the appropriate command from the paint command menu. When you exit, you will be asked whether or not the changes you made while in Paint mode should be placed in the current bitmap. **Paint** is particularly useful for erasing or filling in large regions in bitmaps. See the Window Menu section in Chapter 7 for a detailed discussion of all the paint menu options.



Figure 10-23. **Paint Command** Menu

ShowAsTile

Tesselates the current bitmap in the gray bar. This is useful for determining how a bitmap will look if it were made the display background. The tiled display does not automatically change as the bitmap changes. To update it, use the ShowAsTile command again.

Grid On/Off

Turns the editing grid display on or off.

GridSize

Allows you to specify the size of the editing grid. When you select this option, a number menu appears, giving you a choice of several point sizes for the grid (see Figure 10-24).

3	7	20
4	8	24
5	12	28
6	16	32

Figure 10-24. **Number** Menu

When you select a size, the editing portion of the bitmap editor is redrawn. A smaller size allows you to edit more of the bitmap without scrolling; a larger size makes it easier to turn individual bits on and off. The original size is chosen heuristically. It is typically about 8. Click outside the number menu to abort this operation.

Reset

Sets all or part of the bitmap to the contents it had when you originally called the bitmap editor. When you select this option, a second menu provides a choice between resetting the entire bitmap or just the portion that is in the edit area (see Figure 10-25).



Figure 10-25. **RESET how much?** Menu

This second menu also acts as a confirmation, since clicking outside of this menu results in no action being taken. Note that if the entire bitmap appears in the edit area, the menu only has the **WholeBitmap** option.

Clear

Sets all or part of the bitmap to white space. As with the Reset command, a second menu provides a choice between clearing the entire bitmap or just the portion that is in the edit area (see Figure 10-26).



Figure 10-26. **CLEAR how much?** Menu

Cursor

Sets the cursor to the contents of the lower left part of the bitmap. This operation next prompts you to specify the new cursor's active pixel. You do this by clicking somewhere in the lower left 16 x 16 portion of the grid. Cursors created this way are typically very short lived.

Note: This option is intended for people extending the NoteCards environment. We recommend that non-programmers do not use this option.

OK

Copies the edited bitmap image into the original bitmap, exits the bitmap editor, and closes its edit window. The image you modify using the editor is a copy of the original bitmap. Unless you exit the bitmap editor via **OK**, no changes are made to the original bitmap.

Abort

Exits the bitmap editor without making any changes to the original bitmap. Contrast with **OK**.

[This page intentionally left blank]

11. SYSTEM CARDS

Cards can be broadly divided into two categories. System cards are cards where the contents are built by the system for you. User cards are those cards for which you create the contents. This chapter discusses system cards.

This chapter explains how to use:

- Search cards
- LinkIndex cards
- Document cards
- Browser cards

Search Cards

Search cards search through all boxes and cards looking for titles containing a specified string of characters. A new card is created containing links to these cards. To create a search card, point to New Cards entry in the banner with the middle button. Then slide the mouse to the Search entry, as shown in Figure 11-1.

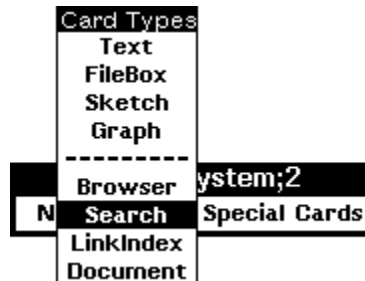


Figure 11-1. **Search Card** Menu Option

The search card also contains the day and time the list was compiled. Search is case sensitive so the string must be typed exactly as it appears in the title. For example, the Search card below was created by answering the prompt for a search string with **Creat**.

Search strings allow wild card characters.

* Matches any number of characters.

? Matches one character.

If you enter a string without wild cards, a "*" will be inserted at each end. If you enter any wild cards, you must enter a "*" at each end of the search string if you want an internal match. For example, a search of "xxawczz" with "a?c" fails, but "*a?c*" succeeds. Note that "awc" would also succeed.

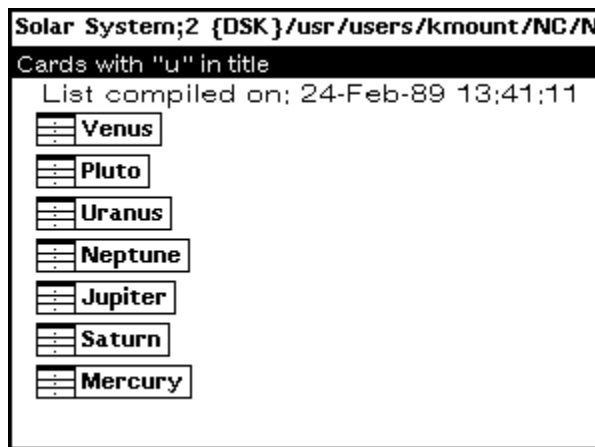


Figure 11-2. Search Card Example

Note: This list is not updated as new cards are entered into the NoteFile. The date and time the list was compiled can be used to help determine if the list in the card is out-of-date.

LinkIndex Cards

LinkIndex cards build sorted lists of all of the cards in the Notefile connected by the specified link type(s). A LinkIndex card helps clarify the link-based relationship taxonomy which has been created within a Notefile. As for search cards, link index cards are created using the middle button menu of the New Cards entry in the notefile banner, as shown in Figure 11-3.

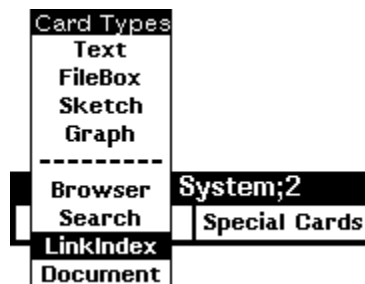


Figure 11-3. LinkIndex Card Menu Option

The following is an example of a Link Index card.

Sample Link Index Notecard
Sorted link index compiled on: 26-Feb-85 15:29:32 for linktypes; Explanation
Introduction to the NoteCards Model <Explanation> TO Model Underlying Notecards
Model Underlying Notecards <Explanation> TO Notes Organization Flow Products Sources

Figure 11-4. LinkIndex Card Example

After creating a link index card, a list of Link Types is displayed from which to select the link type with which the index will be built. Both links to cards/boxes and links from cards/boxes may be selected for tracking. In the example above, Explanation was selected. The prompt window also asks whether back links should be built to the cards or boxes connected by the specified links. Back links are useful if the link index is being used to collect cards for later modification of link types. The example shown above was compiled with the back links option off. Then **Done** was moused.

Link Index Specs?			Include Link Icons?	DONE RESET ABORT
Forward Links	Backward Links			
ListContents	←ListContents	☐	☒ Yes ☐ No	
BrowserContents	←BrowserContents	☐		
FiledCard	←FiledCard	☐		
SubBox	←SubBox	☐		
DocBackPtr	←DocBackPtr	☐		
LinkIndexBackPtr	←LinkIndexBackPtr	☐		
test	←test	☐		
Comment	←Comment	☐		
See	←See	☐		
Unspecified	←Unspecified	☐		
<input type="button" value="ALL"/>	<input type="button" value="ALL"/>			
<input type="button" value="CLEAR"/>	<input type="button" value="CLEAR"/>			

Figure 11-5. Link Index Specs? Menu

After the prompt is answered, link index compiles the set of cards/boxes linked by the chosen link types and displays the list in the Link Index card, sorted alphabetically. This new card also contains the day and time the list was compiled.

Note: This list is not updated as new cards are entered into the NoteFile. The date and time the list was compiled can be used to help determine if the list in the card is out-of-date.

Document Cards

creates a card that contains textual information collected from all descendant cards and/or boxes from a specified FileBox or NoteCard. Once Document is selected (again using the middle button of the **New Cards** entry in the Notefile banner), the user is prompted for a starting box or card. After this selection, a sub-menu of Make Document parameters is displayed.

Make Document Parameters		
Headings From Fileboxes	Titles From NoteCards	Build Back Links
Numbered Bold Unnumbered Bold NONE	Bold Not Bold NONE	Cards and Boxes Cards only Boxes only NONE
Copy Embedded Links		Expand Embedded Links
ALL NONE Select		ALL NONE Select
		DONE RESET ABORT

Figure 11-6. **Make Document Parameters** Menu

These parameters are set to default values displayed to the right of each parameter. It is possible to change a value by placing the cursor over the parameter and depressing the left mouse button. This pops up a menu of values. Select a value from this menu. The parameters and their choice of values are described below.

HeadingsFromFileBoxes

Creates section headings and subheadings for the document according to the option selected.

NumberedBold includes numbered headings, in bold, of FileBox descendants of the originating FileBox. Each heading is numbered in standard section heading form. For example, "1.3" would be assigned to the third sub-filebox of the first sub-filebox of the originating FileBox.

UnnumberedBold includes unnumbered headings, in bold, of FileBox descendants of the originating FileBox.

NONE excludes headings of FileBoxes including the originating FileBox.

TitlesFromNoteCards

Enables the document card to be generated with paragraph-level labeling according to the option selected.

Bold includes titles, in bold, of all card descendants of the originating card.

NotBold includes titles, in regular type, of all card descendants of the originating card.

NONE excludes all titles of cards including the originating card (if starting from a NoteCard).

BuildBacklinks

Builds back links according to the option selected. Back links to cards allow the user to easily retrieve a card for text revision; back links to boxes facilitate structural revision of a document. In the sample **MakeDocument Parameters** menu shown above, the BuildBacklinks parameter is set to ToCards to facilitate the text editing process. When a final version of the document card has been generated, the user may want to set this parameter to NONE to omit the back links from a hardcopy.

Cards and Boxes	builds links to each NoteCard and FileBox descendant of the originating card or box. The link icon, displayed as a small notecard, is inserted next to the title of the card/box the link refers to.
Cards only	includes link icons only to NoteCard descendants of the originating card or box. The link icon, displayed as a small notecard, is inserted next to the title of the card the link refers to.
Boxes only	includes link icons only to FileBox descendants of the originating or box. The link icon, displayed as a small notecard, is inserted next to the title of the box the link refers to.
NONE	back links are not built for the Document card.

CopyEmbeddedLinks

Allows the user to retain or omit embedded links in descendant cards.

ALL	includes any links appearing in descendant NoteCards.
NONE	omits any links appearing in descendant NoteCards.
Select	enables the user to select link types to be included in the Document card. When select is designated, a list of Link Types pops up from which the user selects the link types to include.

ExpandEmbeddedLinks

Allows the user to replace embedded links in descendant cards with their text. Note that cards that appear multiple places in the structure defined by the descendant cards are only expanded once, thus avoiding unnecessary duplication of text. In the sample MakeDocument Parameters menu shown above, the Select option was chosen, then the Capital link type was selected off of the menu listing link types. Therefore, any Capital links encountered while building the document will cause the text of the cards at the other end of the link to be embedded in the document at the link's location.

ALL	for any link appearing in the text of a descendant card, the link is replaced by the text of the card linked to.
NONE	leaves embedded links alone.
Select	expands embedded links selectively, depending on the link type.

Browser Cards

A card that presents a view of cards/boxes and the links between cards/boxes in a specified portion of the current NoteFile. Given a starting card or box and a set of link types, Browser will create a graph structure showing all NoteCards and FileBoxes linked to and/or from the initial card/box by recursively following all links of the types specified.

This graph structure displays both the specified set of cards/boxes, and the links between the cards/boxes in the set. The following is an example of a Browser card:

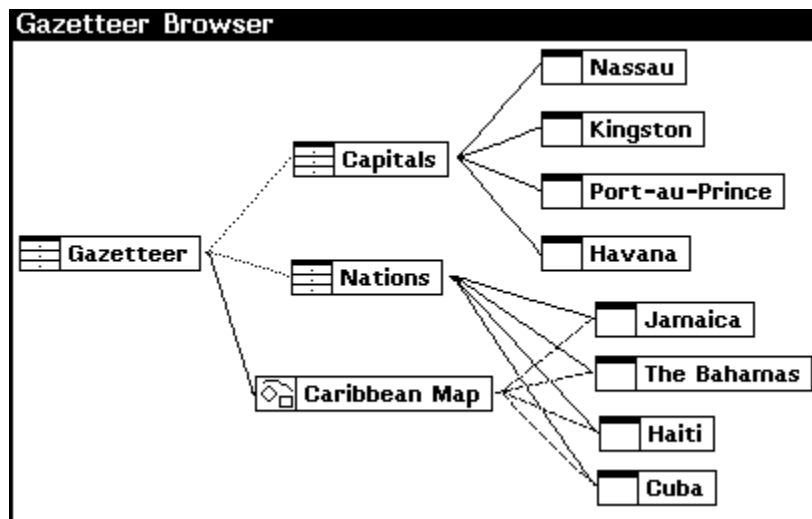


Figure 11-7. Browser Card

Before selecting Browser, the starting card/box or a link icon referring to the card/box must be visible on the screen to allow selection with the mouse. The Browser selection is started also from the **New Cards** entry of the Notefile banner with the middle button. After creating a blank browser card, a prompt window will be displayed above the card asking the user to designate a starting card/box. The user may select the initial card/box or abort the browser by selecting **Cancel**. In the above example, the box chosen was Gazetteer.

After designating the starting card/box, a list of Link Types is displayed from which to select the types of links the browser should follow. Both links to cards/boxes and links from cards/boxes may be selected for tracking. You may select the types and **Done** or abort the browse by selecting **Cancel**. In the above example, the link types chosen were FiledCard, SubBox, and Comment. Once the link types have been determined, the browser compiles the set of cards/boxes and displays it in the Browser card.

Browser Specs?		
Forward Links	Backward Links	Depth
ListContents	←ListContents	0
BrowserContents	←BrowserContents	1
FiledCard	←FiledCard	2
SubBox	←SubBox	3
DocBackPtr	←DocBackPtr	4
LinkIndexBackPtr	←LinkIndexBackPtr	5
test	←test	6
Comment	←Comment	7
See	←See	8
Unspecified	←Unspecified	9
		INF
ALL CLEAR		ALL CLEAR

Format	Orientation	
GRAPH	Horizontal	DONE RESET ABORT
LATTICE	Vertical	
COMPACT	Reverse/Horizontal	
FAST	Reverse/Vertical	

Figure 11-8. Browser Specs? Menu

The links can be drawn in one of two ways, depending on a parameter value set in the **Change Parameters** command from the **Main Menu**. The first method shows the links as solid lines; the second method uses dashed lines. With the second method, each link included in the Browser is assigned a different style of dashing. There are six styles of dashing; therefore, dashing for links after the sixth type is of the same style. The Browser card shown above was created with the `LinkDashingInBrowsers` parameter set to Yes.

A legend is attached to the upper right-hand border of every Browser. This legend displays each type of link present, and, if the links are dashed, the legend displays the particular pattern of dashing. For the Browser card shown above, the legend appears as follows:

Links	
FiledCard	—————
SubBox	
Comment	-----

Figure 11-9. Browser Card Legend

The characteristics of browser cards and graph cards are identical except for the single-item menu that is displayed by pointing to the title bar with the cursor and pressing the middle mouse button. (See the Graph Editing Menu section.) The commands in this menu follows:

Recompute Browser

Recompute Browser modifies the browser card to reflect the current NoteFile starting from the same card or box and following the same link types as originally specified.

Warning: Nodes and links added or deleted through the **Graph Editing Menu** commands **Add Node**, **Add Link**, or **Delete Link** are removed when **Recompute Browser** is executed.

Multiple Roots

Browsers can contain multiple roots, in which case the graph will be laid out as a forest.

Dashed Links

Dashed browser links should be used carefully because of speed considerations. There are currently nine different dashing styles possible. If a browser contains instances of more than nine different link types, then the last dashing style will be used repeatedly for each link type beyond the ninth. Link dashing is a user-settable option in the GlobalParameters menu (see Section #). In Figure # the links legend is attached to the upper right corner of the browser.

Arrowheads

Arrowheads can be drawn on browser links. These show the direction of the notecards link being represented in the browser. This is a user-settable parameter in the GlobalParameters menu with possible values AtMidpoint, AtEndpoint, or None. If AtMidpoint or AtEndpoint is specified, then arrowheads will be drawn at link midpoints or endpoints, respectively. However, in either case, if two browser nodes are connected by more than one link, then any arrowheads for those links will appear at the midpoints (so as not to overlap).

Browser Specs

Link types is one of a number of browser specs, as already mentioned. Also included are browser depth, format, and orientation. These are accessible through a BrowserSpecs stylesheet, a collection of five menus. The browser specs stylesheet is shown in Figure #. For general details on the stylesheet interface see Section #. In this case, the forward and backward link types menus are multi-selectable, that is, more than one entry can be chosen. The other three menus are used to make single selections.

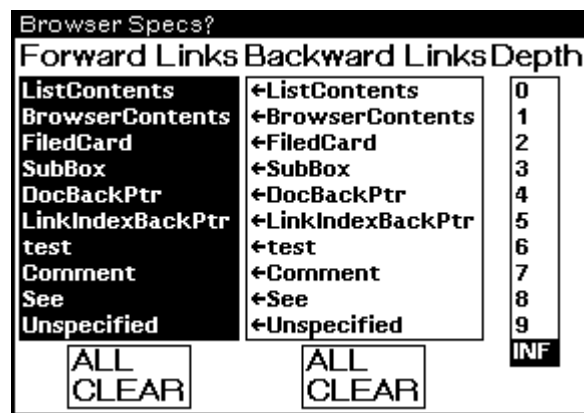


Figure 11-10. Link Types Selection Section of **Browser Specs?** Menu

Forward and backward link types function as explained in the chapter on Links. That is, the browser will contain only nodes for cards reachable from the root cards by following forward links in "line of direction" or backward links in "reverse line of direction." Browser depth is chosen from a menu containing entries for the integers 0 through 9 and INF (or infinite depth). The default is INF, meaning that the browser will not be cut off until there are no more links to follow from leaf nodes. Choosing depth 0 means that only the root nodes will appear (and no links).

Browser format is one of *GRAPH*, LATTICE, COMPACT, or FAST. The latter three are provided by the grapher package and correspond to lattice, compact forest and fast forest, respectively. COMPACT and FAST generate virtual nodes (in double boxes) whenever two or more links would be drawn to the same node. LATTICE only generates virtual nodes when a cycle exists in the graph. *GRAPH* is a format that never generates virtual nodes. The drawback to using *GRAPH* is that a cycle can cause lines to be drawn that cross boxes or overlap other lines. Thus you may have to move nodes around for legibility after computing the browser. The default is LATTICE.

Browser orientation is one of Horizontal, Vertical, Reverse/Horizontal, or Reverse/Vertical. These specify whether the graph is layed out left-to-right, top-to-bottom, right-to-left, or bottom-to-top, respectively. The default is Horizontal.

Middle Button Title Bar Menu Options

The options to the middle button menu invoked from a browser's title bar are RecomputeBrowser, RelayoutGraph, ReconnectNodes, UnconnectNodes, ExpandBrowserNode, GraphEditMenu, ChangeBrowserSpecs, ChangeBrowserRoots, BrowserOverviewWin and ChangeOverviewSpecs.

RecomputeBrowser causes the current contents of the browser to be thrown away and recomputed. You can optionally specify a new set of root nodes.

RelayoutGraph does not rebuild the graph, but rather causes the nodes and links of the graph to be repositioned on the screen. This will destroy any work you have done moving nodes within the graph.

ReconnectNodes first causes any link edges in the graph to be erased. (Note, however, that non-link edges, those created by "AddEdge" as described below, are ignored.) Then, each node in the graph is connected to every other node in the graph for which there is a link between them having one of the currently selected link types. This can be useful for several reasons:

1. When the linking structure between cards has changed, but the current browser layout needs to be preserved.

2. When some browser nodes need to be moved, but dragging the connected links is too slow. In this case, do `UnconnectNodes` followed by `ReconnectNodes` (after you've moved the nodes around).
3. When special browser layouts are desired. For example, suppose you like the layout that Grapher gives you when certain links are left out or when you limit the depth. Then calling `ReconnectNodes` will fill in the missing links without affecting the graph's layout.

UnconnectNodes erases all edges in the browser. This is useful for positioning a browser's nodes before invoking `ReconnectNodes`.

ExpandBrowserNode allows you to enlarge the graph under a given node. After selecting a node, you're asked for a depth (defaults to 1). The graph is then expanded under the selected node to the given depth, following any currently selected links. Note that `ExpandBrowserNode` calls `LAYOUTGRAPH` so any existing special node arrangements will be lost.

GraphEditMenu brings up the graph editing menu. See the description below.

ChangeBrowserSpecs brings up the `BrowserSpecs` stylesheet to allow you to change any of the browser specs. These changes will be noticed at the next `RecomputeBrowser`, `ExpandBrowserNode`, etc.

ChangeBrowserRoots allows to change the roots for the browser card. These changes are noticeable at the next `RecomputeBrowser`, `ExpandBrowserNode`, etc.

BrowserOverviewWin causes an overview window to be created and attached to the upper left corner of the Browser. If this Browser has never had an overview window, then the overview window has the default height and width. If there previously was an overview window that has since been closed, then its height and width are reused.

ChangeOverviewSpecs brings up the Browser overview stylesheet. The stylesheet contains three submenus. The first two specify where to attach the overview window and the third specifies the mode of operation.

Editing the Browser Manually and "Structure Editing"

Browsers are first useful as static views of a portion of a notefile. But, they can also serve as "reach-through" ports, allowing changes to be made to the notefile. This capability is handy when you want to create structure, but not content. For example, when brainstorming ideas for a paper, you might have card titles and links in mind, but want to put off the job of filling in each card's text till later.

A browser consists of **nodes** and connections (or lines). These usually correspond to cards and links in the notefile. However, it is also possible to annotate a browser with **labels** which are pieces of text representing nothing in the notefile (at least not as far as `NoteCards` is concerned). Edges can also be created joining labels and/or normal nodes which do not correspond to links in the notefile. These are called **non-link edges**.

Creation and deletion of nodes and edges in the browser is accomplished using the `GraphEditMenu`. (In Figure #, this menu is attached to the lower right corner of the browser.) This menu can be obtained either by right-buttoning in the browser window or by choosing `GraphEditMenu` from the title bar middle button menu.

The options in the `GraphEditMenu` can be used to create either nodes representing existing cards in the notefile, new nodes and corresponding cards, or browser labels with no corresponding card. There are also operations for creating browser edges and

accompanying links as well as edges without notefile analogues. Edges and nodes can be deleted in two ways: either such that the corresponding link or card is deleted as well as the node or edge, or just such that the node or edge is removed from the browser. These options are described further below.

Create Card causes a new card to be created in the current Notefile and a corresponding node for it to be included in the browser. You're asked for the type of the new card, its title, and where to position the node representing it.

Insert Card adds an existing card to the browser. You are asked to point to a card (title bar or link icon) on the screen that this node is to represent and then to position the node.

Delete Card causes a card to be deleted and its corresponding node in the browser to be removed. You are asked first to choose the node representing the card to be deleted and then to confirm the removal of the node (type "y" to confirm) and the deletion of the card. If the selected node is one of a set of virtual nodes (double boxed), then all nodes in the set (i.e. representing the given card) are removed.

Create Link causes a new link to be created between two existing cards and a corresponding edge to be drawn in the browser. (We call such an edge representing a Notecards link, a "link edge." See AddEdge below for creating non-link edges.) You're asked for the "From" and "To" nodes in the browser corresponding to the cards to be linked as well as a link type. The link icon for the new link is positioned at the cursor point in the From card if the card has text substance and an open window. Text cards with closed windows have links inserted at the start of the text stream. Otherwise, the new link is a global link. You can have multiple link edges between pairs of cards. In this case the edges are displayed in a spline or "flower" arrangement.

Delete Link causes a link in the Notefile to be deleted and the corresponding edge in the browser to be removed. You first pick the "From" and "To" nodes corresponding to the source and destination ends of the link respectively. Then, if there is only one link between those two cards, the link is deleted after user confirms. If there are multiple links between the two cards, then the user chooses from a menu of link types.

Move Node allows you to change the position of any node, rubber banding any edges pointing to it. You're asked to point to the node by left-buttoning, and holding down the left button, drag the node to its new position.

Remove Node removes a node from the browser. It does not delete the card (if any) that the node represents. Edges into and out of the node are also removed. If the selected node is one of a set of virtual nodes representing the same card, then you will be told how many nodes will be removed with this one and will be asked to confirm. The only way to remove only one node of a set of virtual nodes, is to first manually remove edges into and out of it using RemoveEdge. Then RemoveNode can be used to remove only the one virtual node.

Connect Nodes draws a line between two nodes in the browser. This edge does not correspond to a real link in the Notefile. To avoid confusion, it is best to have the arrowheads option on (see Section #) in this case, since edges formed by AddEdge do not have arrowheads (or dashing). Only one such edge is allowed between any two nodes

and none if there are already link edges between the nodes. Thus doing CreateLink&Edge will remove any existing non-link edge.

Disconnect Nodes removes an edge from the browser. It does not delete the link (if any) that the edge represents. The user is asked to select the "From" and "To" nodes of the edge.

Add Label puts a "label node" into the browser that does not represent a Notecard. You are prompted for a string forming the node's label and then must position the label node. This node is not boxed. (But note that "virtual" label nodes can be boxed and thus can be confused with non-virtual regular nodes.)

Smaller Label is used to decrease the font size of label nodes. Note that it does not work for regular non-label nodes.

Larger Label is used to increase the font size of label nodes.

Toggle Shade toggles the shade of a node between black-on-white and white-on-black. This can only be performed on label nodes (not on nodes representing Notecards).

FIX MENU causes the GraphEditMenu to be affixed to the lower right edge of the browser window. Note that this does not prevent you from obtaining the menu via right button inside the window.

Overview Windows on Browser Cards

A Browser overview is a small window attached to a Browser that contains a shrunken view of the entire Browser graph (not just what currently appears in the Browser's window). Within this overview is a wire frame representing that portion of the Browser graph that is currently displayed in the Browser window. The wire frame can be moved in the overview window to effect large scale scrolling in the Browser window. Likewise, scrolling the Browser window moves the wire frame in the overview window.

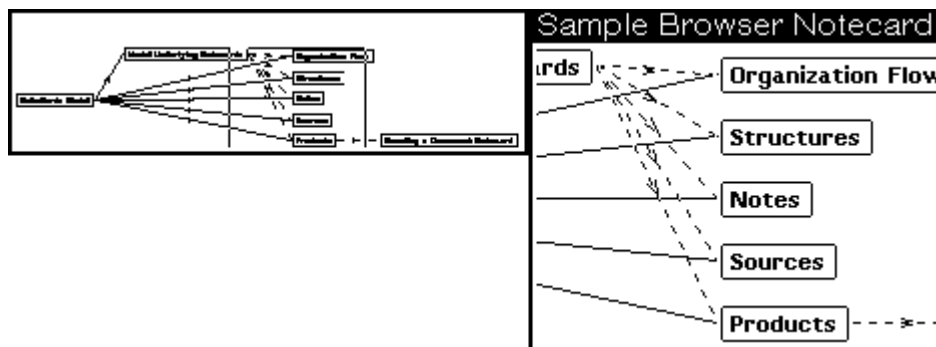


Figure 11-11. Browser Card with Overview Window

The details of creating and manipulating overview windows follow.

Creating an Overview Window

There is a menu item **Browser Overview Win** on the Browser middle button menu. Clicking on this causes an overview window to be created and attached to the upper left

corner of the Browser. If this Browser has never had an overview window, then the overview window will have the default height and width. If there previously was an overview window that has since been closed, then its height and width will be reused.

Reshaping the Overview Window

If you reshape the overview window, then its contents are expanded or shrunk to "fill the container," and then the overview window is reattached to the Browser.

Scrolling and the Wire Frame

In the overview window there is a wire frame whose location and size in the overview match the scrolled position of the Browser window relative to the entire graph. In effect, the wire frame provides a 2-dimensional scroll bar. Its position and size are dynamically updated as the Browser window is scrolled and reshaped.

Furthermore, the wire frame can be moved in the overview window to effect large scale scrolling of the Browser window. This is done by clicking the mouse inside the overview window. The cursor is "pulled" to the nearest corner of the wire frame, and as long as the button is held down, you can drag the wire frame to a new position. If the button is let up with the wire frame outside the overview window, then the scrolling operation is cancelled.

Recomputing the Overview Contents

For large graphs, the operation of creating the overview contents can be expensive. Thus if the Browser's contents change, the overview window doesn't update automatically. In order to recompute the overview window based on current Browser contents, hit Redisplay from the right button menu of the overview window.

Browser Overview Stylesheet

Certain parameters governing the operation of Browser overviews can be changed via the Browser overview stylesheet. To bring up the Browser overview stylesheet, click on **Change Overview Specs**, from the Browser title bar middle button menu. The stylesheet contains three submenus. The first two specify where to attach the overview window and the third specifies the mode of operation.

The position at which the Browser overview is attached can be selected by specifying values from two menus: the edge along which to attach (top, bottom, right, left), and the position along that edge (top/right, center, left/bottom). [Currently center position only works for top and bottom edges.]

The third submenu lets you select the mode. This can have one of three values:

Compress Overview Win: This forces the overview window to be just big enough to fit the overview contents plus the wire frame. This means that when you reshape, the window will be compressed along one dimension to fit the overview.

Expand Overview: This doesn't change the window size, but rather increases the size of the overview graph to fit the window shape you choose. This is done by scaling up in the dimension needed to fill the window. (Scaling is only done to node position, NOT node size. So what you're really getting is stretching out of your links.)

Neither: The old style. That is, the window is whatever shape you make it and the overview contents are in the same aspect ratio as Browser contents. This is the default.

Tidbits

When the ratio between overview window size and graph size gets smaller than a threshold value (currently .3), nodes in the overview switch from being shrunk but readable to being unreadable shaded boxes.

Use the wire frame in the Browser overview to scroll the Browser beyond its displayed region. In fact, you can even pull the wire frame partially outside the overview window and the Browser will scroll appropriately. (Note that if you do such a thing, you may want to put a label in that new area or move a node over there or something to make your next Redisplay of the overview window show the increased area.) As mentioned above, if you pull the wire frame totally outside the overview window, you've done a no-op.

[This page intentionally left blank]

12. MENUBOX ICON

The MenuBox Icon, illustrated in Figure 12-1, is the main interface to NoteCards. Through this icon you access all the system NoteCards menus.

This chapter explains how to:

- Create notefiles
- Perform basic operations on notefiles
- Recover notefiles when they are damaged
- Perform structure operations on notefiles



Figure 12-1. MenuBox Icon

Notefile Options

It is through the Notefile-Operations menu that you interact with notefiles. The word "Operations" is usually abbreviated as "Ops" and, as a result, this menu is usually referred to as the **Notefile Ops** menu.

Access the **Notefile Ops** menu by holding the left mouse button down in the **Notefile** option of the MenuBox Icon (see Figure 12-2). The menu has three regions, separated by dashed lines. The top region contains operations you can only apply to closed notefiles. The middle region contains commands you can only apply to open notefiles. The bottom item, NC FileBrowser, does not share these limitations. It provides you with a notefile interface.

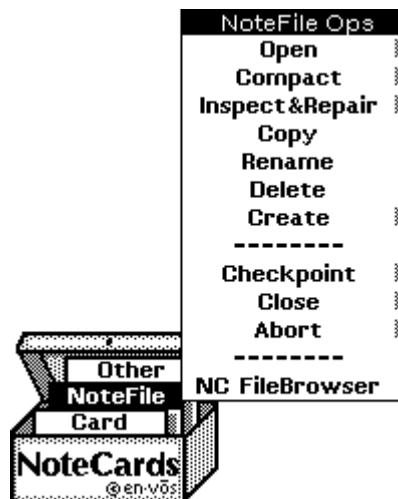


Figure 12-2. MenuBox Icon's Notefile Ops Menu

Open

Opens an existing notefile. If you have not previously opened any files, the system prompts you to type in a file name. The file name is not case sensitive. If you provide just a file name with no path, the system looks in the connected directory for the notefile. If NoteCards does not find the file there, it asks if you want to create the file.

The system maintains a list of all notefiles you open. Subsequently, when you select **Open**, these files are presented to you in menu format. If the file you want to open is not on the menu, select the **--Other Notefile--** option.

There are two submenu options for Open, **Open Read/Write** and **Open Read-only** (see Figure 12-3).



Figure 12-3. **Open** Submenu

Open Read/Write

Is the same as selecting **Open** from the top-level menu.

Open Read-only

Opens the file but does not allow you to make any changes to the file. When you open a file read-only, the filename appears in its Banner with a read-only prefix (RO:) (see Figure 12-4).

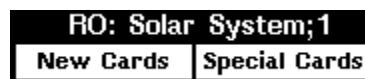


Figure 12-4. Notefile Opened Read-Only (Banner shows RO: Prefix)

If the notefile is damaged, the menu shown in Figure 12-5 is displayed.

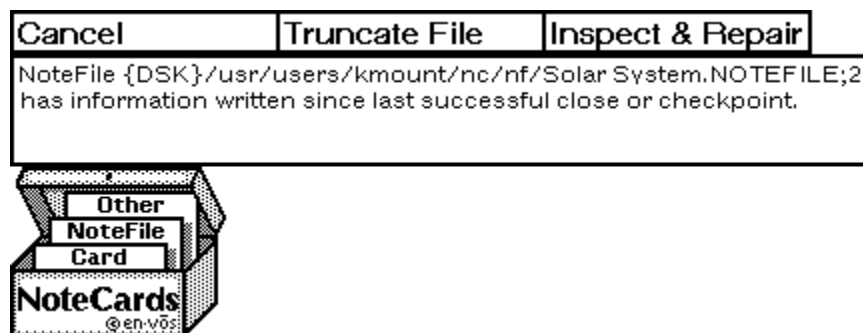


Figure 12-5. Menu Displayed when Opening Damaged Notefile

Cancel causes the open notefile operation to be aborted.

Truncate File deletes everything after the checkpoint pointer.

Inspect & Repair causes the notefile inspector to be called on your notefile. This is an easy, but slow, way to incorporate these post-checkpoint changes into the notefile. For more details, see the Inspect & Repair section below and Appendices A and C.

Compact

Compresses a notefile by deleting old information. NoteCards never actually overwrites any information in the data area of a notefile. When cards are revised, old information remains in the notefile. Compacting creates a new, compressed version of the notefile by copying into the new version only the latest information from the old version. To understand the structure of notefiles and compacting, see Appendix A, .

Compacting can take 5 to 20 minutes, depending on the size of the notefile.

There are two submenu options off **Compact**: **Compact To New File** and **Compact in Place** (see Figure 12-6). Compact To New File is the default.



Figure 12-6. **Compact** Submenu

Compact To New File

Is the same as selecting **Compact** from the main menu. NoteCards prompts you to type in a new file name (see Figure 12-7). The default is the same file name with a higher version number (see Figure 12-8).

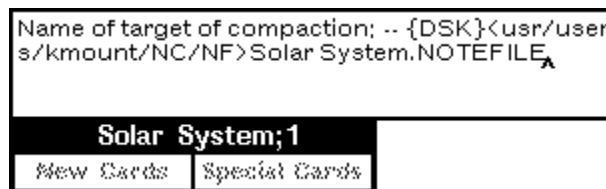


Figure 12-7. NoteCards Prompt for Compact Operation Output File Name



Figure 12-8. Notefile Banner after Compacting to Same File name (old version is "Solar System;1."; new version is "Solar System;2.")

Compact In Place

Does not build a new copy of your notefile. The contents of the old file are rearranged in such a way that old versions of cards are written over. This is useful when your notefile is large and there isn't room to store another version. However, be aware that when you use this option you are not able to back up cards to previous versions by using the **Inspect & Repair** utility. We recommend compacting to a target file and saving the uncompactd version until you are sure that the older versions of cards are not needed.

Inspect & Repair

Reads the data area of your notefile, looking for good card parts, including outdated and deleted versions. It then allows you to delete or backup card parts to earlier versions through its menu interface. Only when the notefile is considered healthy are you allowed to perform the link rebuilding.

Inspect & Repair has two submenu options, **From Links** and **From Contents**, illustrated in Figure 12-9.

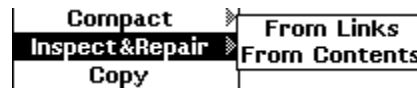


Figure 12-9. **Inspect & Repair** Submenu

From Links

Rebuilds the notefile index using only the links part of the notefile.

From Contents

Rebuilds the notefile index using the links part of the notefile as well as the links stored in the bodies of individual note cards. **From Contents** takes longer to rebuild a notefile index.

For a detailed explanation of the **Inspect & Repair** utility, see Appendix C.

Copy

Allows you to copy a notefile.

Prompts for a new name and path for the notefile to be copied and copies it. This is the same as doing a file copy.

Rename

Allows you to rename a notefile.

Prompts for a new name and path for the notefile to be renamed and renames it. This is the same as executing a file rename.

Delete

Deletes a notefile.

Enter Y and press Return to confirm the deletion, or press Return to cancel the deletion. If you enter Y, a second confirmation is required to ensure against unintentional deletion.

Create

Creates a new empty notefile. When you use **Create**, it is possible to assign an already existing name to the new notefile, thereby creating more than one version of the notefile.

Create has two submenu options, **Create without Open** and **Create and Open** (see Figure 12-10).



Figure 12-10. **Create** Submenu

Create without Open

Is the default action. It prompts you for a new notefile name, creates a new notefile, and place the new notefile's Banner on the screen. If you supply only a name and no path, the notefile is created in your connected directory.

Create and Open

Does all of the above, and opens the new notefile.

Checkpoint

Saves the contents of any active cards to the notefile without closing the cards or the notefile. The index array is written out to the file and the checkpoint pointer is reset to the end of the file.

A notefile consists of two parts, an index area and a data area. For each card, the index contains the following four pointers into the data area: notecard's contents, title, property list, and list of links. For example, when you change a card's title, NoteCards writes the new title at the end of the file, and the card's title pointer is changed to point to the new title.

To increase access speed, index modifications are written in memory and not to the file till checkpoint, or close time. At close or checkpoint update, a checkpoint pointer is updated to point to the end of file. New data, such as a new title, is still written to the file, but always at the end of the file. Thus if a crash occurs, due to a power failure or some other mishap, and later you reopen the notefile, NoteCards can notice the extra data beyond the checkpoint pointer.

Each time a notefile is opened, the checkpoint pointer is compared with the end of file pointer. If they don't agree, an error message is displayed. At this point, you must select one of three options from a menu: **Cancel**, **Truncate File**, or **Inspect & Repair** (see Figure 12-11).

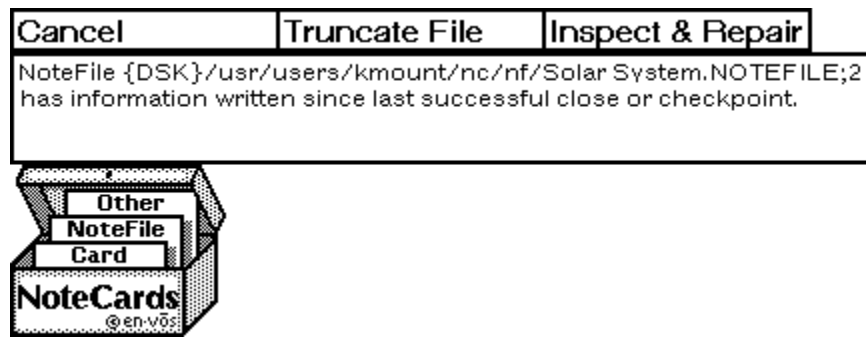
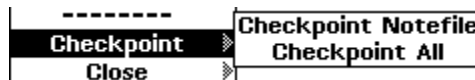


Figure 12-11. Menu which appears when damaged notefile is opened

Cancel causes the open notefile operation to be aborted.

Truncate File deletes everything after the checkpoint pointer.

Inspect & Repair causes the notefile inspector to be called on your notefile. This is an easy, but slow, way to incorporate these post-checkpoint changes into the notefile. For more details, see Appendices A and C.

Figure 12-12. **Checkpoint** Submenu

Checkpoint and **Checkpoint Notefile** pop up a menu of known notefiles which you can checkpoint (see Figure 12-13).

NoteFiles	
Solar System.;2	{disk}/usr/users/kmount/nc/nf/solar system,notefile;2
o Solar System.;4	{disk}/usr/users/kmount/nc/nf/solar system,notefile;4
o demo.;1	{disk}/usr/users/kmount/nc/nf/demo,notefile;1
test.;1	{disk}/usr/users/kmount/nc/nf/test,notefile;1

Figure 12-13. Menu Known Notefiles

The grayed over notefiles are closed and cannot be checkpointed. Selecting one of the open notefiles causes that notefile to be checkpointed.

Checkpoint All checkpoints all open notefiles.

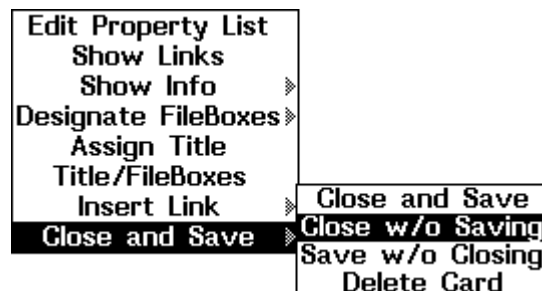


Figure 12-14. Card Menu with the Close w/o Saving Selected

In contrast to **Checkpoint**, when you select **Close and Save** or **Save w/o Closing** from the card menu, the card's contents are written to the notefile but the notefile index is not written to the notefile. If you are anticipating a crash, you should checkpoint your notefile frequently.

Close

Allows you to close a notefile (see Figure 12-15).

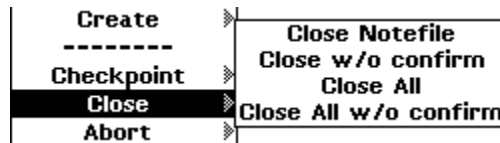


Figure 12-15. **Close** Submenu

Close and Close Notefile

Pops up a menu of known notefiles like that shown in Figure 12-13. Select one of these notefiles to close that notefile. The grayed over notefiles are already closed. Selecting one of these has no effect.

Close w/o confirm

Closes a notefile without asking you about open cards on the screen. Open cards are closed and saved to either their designated FileBox or the "To Be Filed" FileBox.

Close All

Closes all open notefiles.

Close All w/o confirm

Closes all open notefiles without asking you about open cards on the screen. Open cards are closed and saved to either their designated FileBox or the "To Be Filed" FileBox.

Abort

Truncates and closes a notefile (see Figure 12-16).

If you don't want to keep changes made to a notefile, select this option to truncate your file. Do this only if you feel the post-checkpoint work is not worth saving.



Figure 12-16. **Abort** Submenu

Abort and Abort Notefile

Pops up a menu of known notefiles like that shown in Figure 12-13. Select one of these notefiles to abort that notefile. The grayed over notefiles are already closed. Selecting one of these has no effect.

Abort All

Aborts all open notefiles.

NC FileBrowser

Opens up a FileBrowser specialized for NoteCards (see Figure 12-17).

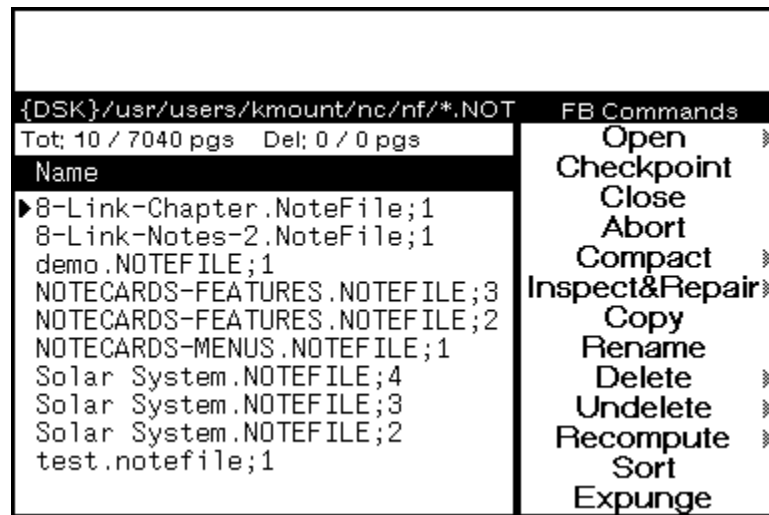


Figure 12-17. NC FileBrowser

Refer to the appropriate sections in this chapter for more information about the **Open**, **Checkpoint**, **Close**, **Abort**, **Compact**, and **Inspect & Repair** commands.

Refer to the appropriate sections in Chapter 14 for more information about the **Copy**, **Rename**, **Delete**, **Undelete**, **Recompute**, **Sort**, and **Expunge** commands.

Card Options

The Card options allow you to perform a **Close**, **Delete**, **Copy**, or **Move** on a series of cards or a card structure (see Figure 12-18).

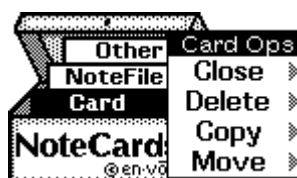


Figure 12-18. Card Ops Menu

Close

Enables you to close any number of currently open cards by simply shift-selecting each one's card-ID into the **Selecting Cards** prompt window.

After you have specified your choices, select **Done** from the menu on top of the Selecting Cards prompt window, or type a carriage return. **Cancel** aborts this command. **Undo** removes the last selected card from the list.

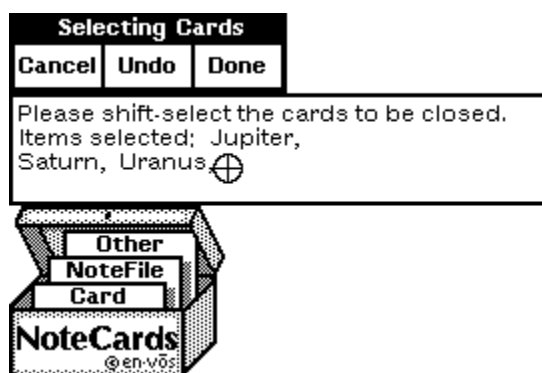


Figure 12-19. Selecting Cards Prompt Window and Menu

The **Close** option has two submenu options.

Close and Close NoteCards

Perform the same action as **Close**.

Close Structure

Closes a linked list of cards. This operation follows both local and global links.



Figure 12-20. Close Submenu

After you select the cards from which to start the **Close Structure** operation, a menu opens asking you what types of links you want to follow (see Figure 12-21). When you are done, select **Done**. **Reset** sets the menu to the values it had when it came up. **Abort**, terminates the operation.

Include cards at:			
Forward Links	Backward Links	Depth	
Comment	← Comment	0	DONE
See	← See	1	RESET
Unspecified	← Unspecified	2	ABORT
ListContents	← ListContents	3	
BrowserContents	← BrowserContents	4	
FiledCard	← FiledCard	5	
SubBox	← SubBox	6	
DocBackPtr	← DocBackPtr	7	
LinkIndexBackPtr	← LinkIndexBackPtr	8	
		9	
		INF	
ALL CLEAR	ALL CLEAR		

Figure 12-21. Include cards at: Menu

Refer to Chapter 8 for an explanation of Forward and Backward links. The **Depth** variable determines how many link levels are traversed in carrying out the operation. If set to a depth of 2, only cards two levels down are closed. For example, if you had cards 1, 2, 3, and 4 on the screen, each forming a sequence (card 1 only pointing to card 2, and card 2 only pointing to card 3, etc.), then if the depth were set to two and you started the close operation from card 1, only cards 1, 2, and 3 would be closed, because only two levels of links would be traversed. Card 4 would remain open on the screen after the **Close Structure** operation.

Delete

Works just like **Close** (see Figure 12-22).

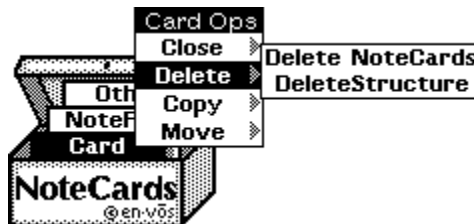
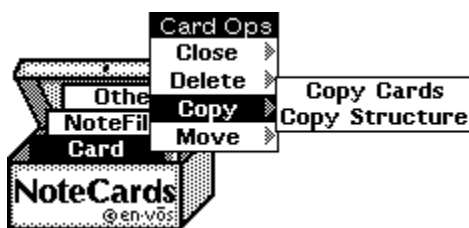


Figure 12-22. Delete Submenu

We caution you to be very careful with the **Delete Structure** command as it is very easy to unintentionally delete large numbers of cards. This command is most useful in conjunction with **Copy Structure** as a way of performing a safe **Move Structure**. First you copy the structure you think you want and then once you have verified that you have gotten the structure you wanted, you use **Delete Structure** to remove it.

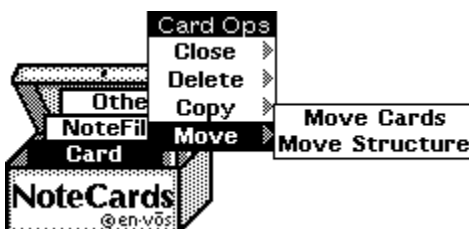
Copy

Works just like **Close**. Allows you to copy a group of cards or a card structure to another FileBox which may be in a different notefile (see Figure 12-23).

Figure 12-23. **Copy** Submenu

Move

Works just like **Close**. Allows you to move a group of cards or a card structure to another FileBox which may be in a different notefile (see Figure 12-24).

Figure 12-24. **Move** Submenu

Other Options

Figure 12-25. **Other Ops** Menu

Edit Parameters

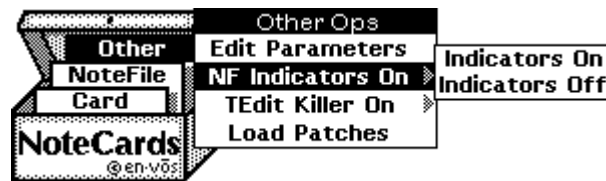
The **Edit Parameters** option brings up the **System Parameters** menu. This menu is discussed in detail in Chapter 13.

NF Indicators On

NF Indicators On and the submenu option **Indicators On** turn on notefile indicators for all the cards on the screen.

Indicators Off

Turns off all notefile indicator windows for all the cards on the screen. Note that whether or not cards come up with notefile indicators is determined by the system parameter **Show Notefile On Cards**. Refer to Chapter 13 for more information about this parameter and changing system parameters.

Figure 12-26. **NF Indicators On** Submenu

Notefile indicators are small windows on the tops of cards which show the name and path of the card's notefile.

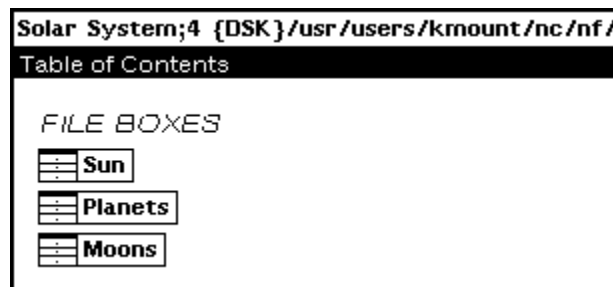
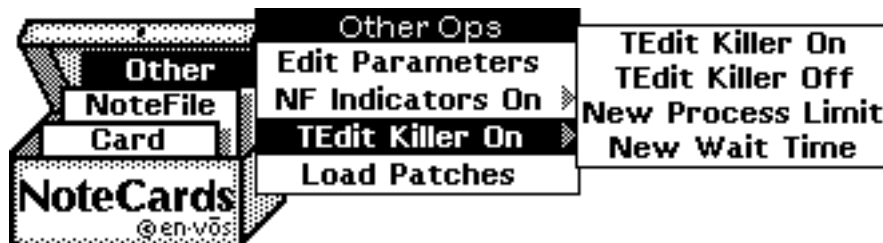


Figure 12-27. Notefile Indicator Window attached to Solar System notefile Table of Contents card from the Solar System notefile

TEdit Killer On

Kills TEdit processes after they have been around for a specified period of time.

Figure 12-28. **TEdit Killer On** Submenu

Since every text-based card and every TEdit window, including TEdit command menus, is a separate TEdit process it is very easy to get 20 or 30 TEdit processes all vying for machine time. This can greatly slow down machine response time. To work with many cards and windows open on the screen at one time, turn the TEdit killer on to help improve response time. Note that TEdit killer only affects text-based cards. The processes for sketch-based cards are unaffected. To restart a TEdit process that has been killed in this manner, click in the TEdit window.

TEdit Killer On

Turns on the TEdit killer process.

TEdit Killer Off

Turns off the TEdit killer process.

New Process Limit

Sets the number of TEdit processes which are allowed to exist. Selecting this option brings up the menu shown in Figure 12-29.

```

Old TEdit
process limit: 5

  0

- AB CLE
  ORT AR
 1 2 3
 4 5 6
 7 8 9
bs 0 ok
  
```

Figure 12-29. TEdit Process Limit Number Pad

A response of 0 allows you to have one TEdit process running and not none.

The default value is 5.

New Wait Time

Sets the number of second which a TEdit process can be idle before it will be killed. Selecting this option brings up the following menu.

```

Current TEdit
killer wait
time: 100 sec

  0

- AB CLE
  ORT AR
 1 2 3
 4 5 6
 7 8 9
bs 0 ok
  
```

Figure 12-30. TEdit Killer Wait Time Number Pad

Note: If both the TEdit process limit and the TEdit killer wait time are set to "0," it can freeze your system.

The default value is 100.

Load Patches

Forces the loading of any patches which were distributed with the NoteCards sysout.

[This page intentionally left blank]

13. SYSTEM PARAMETERS

System parameters allow you to customize the NoteCards environment to your particular needs and tastes.

This chapter explains:

- How to set system parameters
- What each system parameter does

Accessing System Parameters

System parameters are set from the **NoteCards Parameters** menu. To access this menu, hold the left mouse button down on the **Other** option on the MenuBox Icon. When the **Other Ops** menu comes up, still holding the mouse button down, slide off and select the **Edit Parameters** option. Then release the mouse button. You can also execute a single rapid left-button click on the Other menu option and the **Other Ops** menu will come up and stay up. To select the option you want, click on it.

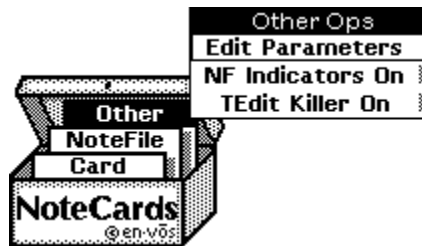


Figure 13-1. MenuBox Icon with the **Edit Parameters** option on the **Other** menu option selected

This action brings up the NoteCards Parameters permanent menu, shown below in Figure 13-2.

Each parameter is preset to a default value. Placing the cursor over the name of the parameter and pressing the left mouse button toggles the value, listed to the right, between Yes and No if binary, and allows selection from an appropriate menu otherwise.

NoteCards Parameters	
NoteCards System Parameters	
Extra TEdit Props	None
Include Card Object In ShowInfo	No
Del TEdit Process When Shrinking	No
Fonts	
Menu Font	(HELVETICA 10 BOLD)
Default Font	(HELVETICA 12 STANDARD)
Link Icon Font	(HELVETICA 10 BOLD)
NoteFile Indicator Font	(HELVETICA 10 BOLD)
NoteFiles	
Show NoteFile On Cards	Yes
New NoteFile Initial Size	1000
Menu Lingers After NoteFile Close	Yes
Show Table of Contents on Open	Yes
Cards	
Force Filing	Yes
Force Titles	No
Default Card Type	Text
Close Cards Off Screen	No
Bring Up Cards At Previous Pos	No
FileBox Cards	
Markers In FileBoxes	Yes
Alphabetized FileBox Children	No
Browser Cards	
Special Browser Specs	No
Arrow Heads In Browsers	None
Link Dashing In Browsers	No
Sketch Cards	
Attach Sketch Menu	No
Links	
Link Icon Border Width	1
Link Icon Multi Line Mode	No
Link Icon Max Width In Pixels	1024
Cross File Link Mode	ASK
Link Icon Show Title Default	Yes
Link Icon Attach Bitmap Default	Yes
Link Icon Show Link Type Default	No
Use Deleted Link Icon Indicators	Yes

Figure 13-2. **NoteCards Parameters Menu**

NoteCards Parameters

Extra TEdit Props

Does not initially appear on the list of parameters for the user version of NoteCards. It is obtained by going into the **Low Level Tools** option on the background menu. This parameter is to be used by people extending the NoteCards environment. Users of NoteCards should not add this parameter to their **NoteCards Parameters** menu.

The value of this parameter is a property list that is appended to the property list passed as the PROPS argument to every call to TEdit. This parameter allows you to set up the initial TEdit props for all new text-based cards.

Selecting this property brings up a type-in window above the parameter editor window. To set the parameter value, type in a sequence of zero or more attribute/value pairs followed by carriage return. The possibilities for attributes and values are described in *A User's Guide to TEdit*, Chapter 8, The Programmer's Interface to Tedit, in the Using the Top-level TEdit Function section.

For example, to set the default line leading to 2 you would type:

```
PARALOOKS (LINELEADING 2)
```

You should be very careful when setting other than LOOKS-related properties. In particular, NoteCards sets properties like QUITFN and PROMPTWINDOW itself, overriding any alternative values for these properties that you may have typed in.

Note: If for some reason, you need to have an expression EVAL'ed in the typein window, precede it with CTRL-Y.

The default value is None.

Include Card Object In ShowInfo

When the value of this parameter is Yes, the Show Info inspector will include the Card Object. This option is for people extending or modifying the NoteCards system.

The default value is No.

Del TEdit Process When Shrinking

If the value of this parameter is "Yes," when you shrink any text-based card, NoteCards will kill the TEdit process behind it. If the value is No, the TEdit process remains running even when the card is shrunk to an icon.

Expanding an icon for a card whose TEdit process has been killed will not automatically restart the TEdit process. Instead, the user has to click the MIDDLE button in the substance of the card, and select the **New Edit Process** item from the singleton menu that appears.

Running TEdit processes use up limited resources. As a result, only about 25 to 30 TEdit processes can be active simultaneously. Setting the value of this parameter to Yes, eliminates unnecessary use of these limited resources when text-based cards are shrunk to icons. The cost is the extra effort needed to restart the process when the card is expanded.

The default value is No.

Font Parameters

When you select any of the parameters in the Fonts section the system displays the **Please select a font:** menu.

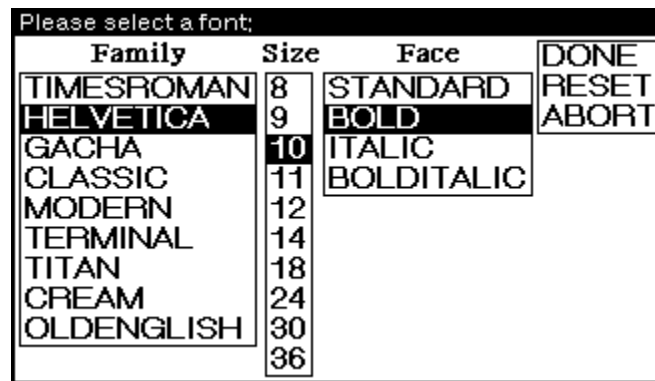


Figure 13-3. **Please select a font** Menu

To change the link icon font, select the values you want for each of the parameters with the mouse and click on **DONE**. **RESET** returns the values to their original settings. **ABORT** returns the values to their original settings and exits.

The value for each font parameter is three-element list which includes the font family, the font size in points and the font face.

Menu Font

The value of this parameter is a description of the font in which the text in all menus will be printed.

The default value is (HELVETICA 10 BOLD) .

Default Font

This parameter controls the font all new text-based cards will use.

The default value is (HELVETICA 12 STANDARD) .

Link Icon Font

The value of this parameter is a description of the font which is used to print the card name and link type that appear inside the links.

The default value is (HELVETICA 10 BOLD) .

Notefile Parameters

Show Notefile On Cards

To associate a card with its notefile, NoteCards provides a notefile indicator window which attaches to the top of the card's title bar (see Figure 13-4).

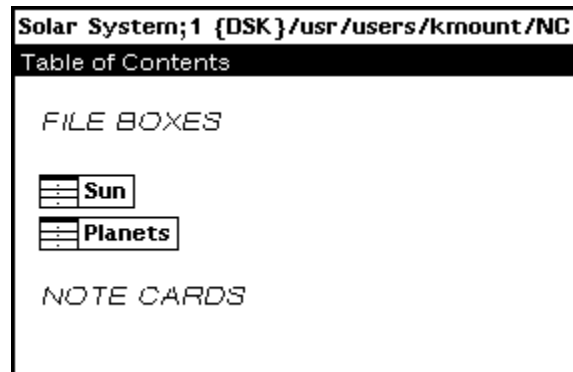


Figure 13-4. Notefile Indicator Window Attached to Top of a Card

The notefile indicator is constructed by concatenating the root file name, the version number, and the full file name. The indicator is left justified. If the card is too narrow to show the entire Indicator string, the full file name is truncated before the root name and version number (see Figure 13-5).

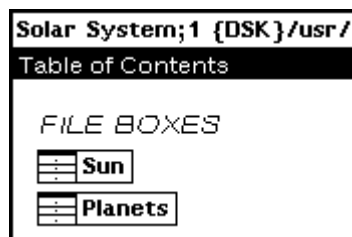


Figure 13-5. Notefile Indicator Window Attached to Top of a Card, with Truncated Full File Name

If the NoteCards system parameter **Show Notefile On Cards** has the value **Yes**, a notefile indicator is attached to every card as it is displayed on the screen.

Occasionally, after you have reshaped a card, a piece of the text in the notefile indicator window is garbled or broken. To redisplay the text in this window, place the point of the mouse cursor in the notefile indicator window, hold down the right mouse button, and select the **Redisplay** option from the window menu which pops up.

The default value is **No**.

New Notefile Initial Size

When you create a new notefile, the system consults the value of this parameter to determine the initial size of the notefile index. The notefile-index size is equal to the number of cards the notefile can hold. Larger notefiles take a little longer to create. Notefiles can be enlarged later on, using the **Inspect & Repair** option.

The default value is 1000.

Menu Lingers After Notefile Close

If the value of this parameter is **No**, the notefile Banner is removed from the screen when the notefile is closed. If the value is set to **Yes**, then the Banner remains on the screen with its menu items greyed.

The default value is **Yes**.

Card Parameters

Force Filing

If you set **Force Filing** to **Yes**, the system requires you to file every card in a FileBox before the card can be closed. Setting the value to **No** eliminates this forced filing, but the system will file the card in the "To Be Filed" FileBox if the card has no parent FileBox when you close it.

The default value is **Yes**.

Force Titles

If you set **Force Titles** to **Yes**, this parameter requires you to title a card before it can be closed. Set the value to **No** to eliminate forced titling. When a card is not titled, it is named "Untitled" by default. This may cause confusion if several cards are left untitled. The system, however, has a unique identifier for each card and will continue to regard them as unambiguous entities.

The default value is **Yes**.

Default Card Type

When you select **the New Cards** option from the notefile Banner with the left mouse button, a card of the default type is immediately created and displayed. This parameter allows you to change this default to suit your needs. Select the new default card type from the list of choices provided.

The default value is **Text**.

Close Cards Off Screen

If set to **Yes**, then when a card is closed, it is first moved off the screen so that the close happens invisibly.

The default value is **No**.

Bring Up Cards At Previous Pos

If set to **Yes**, cards are brought up on the screen in the position they occupied when you last closed them. You are not asked to position a frame image.

The default value is **No**.

FileBox Card Parameters

Markers In FileBoxes

If set to **Yes**, then new fileboxes will contain the markers "FILE BOXES" and "NOTE CARDS" (see Figure 13-6). New child boxes are inserted under the FILE BOXES marker and new child cards under the NOTE CARDS marker. If set to **No**, then new fileboxes come up without markers and new children are inserted at the current cursor position. Note that regardless of the **Markers In FileBoxes** setting, if a filebox has no markers (because you've deleted them) then new children are inserted at the cursor position.

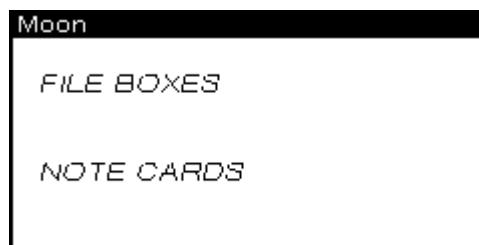


Figure 13-6. FileBox Card showing FILE BOXES and NOTE CARDS markers

The default value is **Yes**.

Alphabetized FileBox Children

If set to **Yes**, then any new cards filed in newly created FileBoxes are alphabetized automatically.

The default value is **No**.

Browser Card Parameters

Special Browser Specs

If set to **Yes**, this parameter causes a sequence of five prompts to appear above the Browser NoteCard during the Browser creation process. These prompts mainly concern the node-link graph properties documented in the Browser Cards section of Chapter 11.

For example, this option may be used to create a vertical browser instead of the default horizontal presentation. Note that a carriage return given as a response to any of the prompts will cause the Special Browser Spec to retain its default value.

The default value is **No**.

Arrow Heads In Browsers

This dictates whether arrow heads are drawn on browser links. The variable can be set to `AtMidpoint`, `AtEndpoint`, or `None`. See the Browser card section of Chapter 11 for details.

The default value is `None`.

Link Dashing In Browser

If set to `Yes`, this parameter builds a Browser with dashed links. Each link in a Browser refers to the type of link being followed. With link dashing, each link included in the Browser is assigned a different style of dashing (up to six styles). A `No` setting means links will be presented as solid lines in a Browser. A Browser with link dashing will take a little longer to create than one with solid lines.

There may be problems printing Browsers with dashed lines.

The default value is `No`.

Sketch Card Parameters

Attach Sketch Menu

If set to `Yes`, the Sketch menu is automatically attached to the right-hand side of each Sketch card when it is opened.

The default value is `No`.

Link Parameters

Link Icon Border Width

Changes the border size of a link icon.



Figure 13-7. A link icon with a border size of 3

The link icon border width can also be set to zero for link icons that blend more smoothly with text.

This is a  **Link Icon** with no border.

Figure 13-8. A link icon with a border size of 10

The default value is 1.

Link Icon Multi Line Mode

This parameter works in conjunction with **Link Icon Max Width in Pixels**. Multi line mode determines whether the link icon will allow multiple text lines or whether all the text will be forced to appear on a single line (see Figure 13-9). If **Link Icon Multi Line Mode** is set to **Yes**, all link icons will display in multiple lines, with each line fitting within the width specified by **Link Icon Max Width in Pixels**. Line breaks occur at word boundaries.

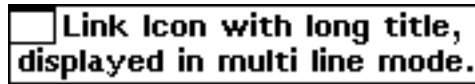


Figure 13-9. Link Icon in Multi Line Mode

The default value is **No**.

Cross File Link Mode

All links in NoteCards are bidirectional or two-way with one possible exception. When you are creating cross-file links, these links can be unidirectional or one-way. A one-way link means that a source card contains a to-link pointing to the destination card, but the destination card does not contain a corresponding from-link pointing back to the source card. This allows you to update a source-card file without being forced to open any corresponding destination-card files.

One-Way	Creates unidirectional links between cards in different notefiles.
Two-Way	Creates the usual bidirectional links between cards in different notefiles.
Ask	The system asks you whether you want One-Way or Two-Way links each time you create a cross-file link.

The default value is **Ask**.

Link Icon Max Width in Pixels

Sets the maximum width in pixels of a link icon. This value includes the card-type bitmap, if it is displayed. All links are truncated at the width specified by this parameter.



Figure 13-10. Truncated Link Icon in Single Line Mode

The default value is 1024 pixels.

Link Icon Show Title Default

This parameter and the following two determine the manner in which link icons are displayed if these parameters are not currently specified in the icon. Which is to say, if the link icon display mode values are set to `System Default`.

If the title field in a link icon is set to "System Default" then the **Link Icon Show Title Default** parameter is consulted to determine whether the title will be shown inside the link icon. If it is set to `Yes`, the title will be shown.

The default value is `Yes`.

Link Icon Attach Bitmap Default

If the bitmap field in a link icon is set to `System Default`, the **Link Icon Attach Bitmap Default** parameter is consulted to determine whether the card-type bitmap is shown next to the link icon. If it is set to `Yes` then the card-type bitmap is shown.

The default value is `Yes`.

Link Icon Show Link Type Default

If the link type field in a link icon is set to "System Default" then the **Link Icon Show Link Type Default** parameter is consulted to determine whether the link type will be shown inside the link icon. If it is set to `Yes` then the link type will be shown.

The default value is `No`.

Use Deleted Link Icon Indicators

If the value of this parameter is `Yes`, when you delete a card, all link icons which point to that card are replaced with a deleted icon (see Figure 13-11).



Figure 13-11. Deleted Icon

If the value of this parameter is `No`, the links to the card are simply removed, leaving no trace behind.

Delete icons are useful when you may be deleting cards where you have link icons pointing to them and these link icons are referred to in the text.

For example, "This link  takes you to the next card in the sequence." If the "Next" card were subsequently deleted with this parameter set to `Yes`, this would become, "This link  takes you to the next card in the sequence." If this parameter were set to `No`, the sentence would become the following more obscure version, "This link takes you to the next card in the sequence." with no visible link.

The default value is `Yes`.

[This page intentionally left blank]

14. FILEBROWSER

The FileBrowser provides a convenient user interface for manipulating files stored on a workstation or file server. It enables you to see, edit, delete, print, load, copy, move, rename, compile, sort, and get several types of information about groups of files.

User Interface

Starting FileBrowser

To open a browser on a set of files select the FileBrowser command from the background menu.

FileBrowser prompts you to create a window by presenting you with a dashed rectangle with the mouse cursor and a small geometric design at the lower right corner.

1. Move your mouse until the upper left corner of the rectangle is where you want it on the screen.
2. Hold down the left mouse button and move your mouse down and to the right, thus expanding the window diagonally, until the window is the right size.
3. Release the mouse button. This creates a window group on your screen in the outlined area.

Next, FileBrowser prompts you for a file pattern. Type a pattern, as described in the section Specifying What Files to Browse, below.

FileBrowser enumerates the set of files matching the pattern you requested. While the enumeration is in progress, the **Recompute** command is grayed out. When the enumeration is finished, you may select files and issue commands. You can scroll the window at any time, even while the browser is busy.

If FileBrowser can't find any files matching the pattern you specified, or you decide you specified the wrong pattern and want to try again, you can specify a new file name pattern from within the browser using the **New Pattern** command; see **Recompute** in the FileBrowser Commands section below.

You can have as many active FileBrowsers open at once as you like.

Specifying What Files to Browse

A full file name in NoteCards consists of a device or host (such as your local disk, a file server), a principal directory and zero or more subdirectories, a file name (possibly including an extension), and a version number. These fields are put together in the form :

```
{HOST}<DIRECTORY>SUBDIRECTORY>FILENAME.EXTENSION;VERSION
```

A file name pattern, as specified to FileBrowser, consists of a file name with one or more pieces omitted or filled with wild cards (*). All the files matching the pattern are listed by FileBrowser. You can browse all the files in a particular directory, all the files in a subdirectory of that directory, all the files in a directory with a particular extension, and so forth. The wild card * can be used to stand for zero or more consecutive characters in the file name. You can use as many wild cards in a pattern as you wish.

If you leave out some of the fields in a file name pattern, the missing fields are defaulted by the system. Omitted fields in the front of the pattern, i.e., the host, device, or directory fields, are filled in by consulting your connected directory. Other omitted fields are filled in with wild cards unless they are explicitly omitted; i.e., the field is empty, but the preceding punctuation is still present. In more detail, some of the cases are as follows:

If you leave out the name of the host/device, specifying *<DIRECTORY>FILENAME*, FileBrowser uses the name of the host/device for the directory to which you are currently connected.

If you leave out both the device and directory names, specifying *FILENAME*, FileBrowser uses the device and directory to which you are currently connected.

If you do not specify a file name, FileBrowser lists all the files in the specified directory (or the connected directory if you also omitted the host and directory).

If you leave out the extension of a file name, FileBrowser lists all the files with the specified file name and any extension. If you omit the extension but include the period that usually precedes the extension, FileBrowser lists only the files with the specified name and *no* extension.

If you omit the version number of the file name, FileBrowser lists all versions of the matching files. If you omit the version number but include the semicolon that usually precedes the version, FileBrowser lists only the highest version of the matching files.

Thus, the minimal pattern you can type is * (asterisk—enumerate all files in the connected directory) or ; (semicolon—enumerate just the highest version of all files). If you press the RETURN key without giving a pattern, FileBrowser aborts the prompt for a pattern, leaving you with an empty browser in which the only things you can do are change some FileBrowser parameters (see the subcommands of **Recompute** in the FileBrowser Commands section below) and then use the **Recompute** command to be prompted for a pattern again.

Examples

The pattern * specifies all files in the connected directory. It is equivalent to *.* or *.*.*.

The pattern <NoteCards>Demo specifies all files in directory NoteCards with name Demo and any extension. It is equivalent to <NoteCards>Demo.*.*.

The pattern <NoteCards>Demo. specifies all files in directory NoteCards with name Demo and *no* extension. It is equivalent to <NoteCards>Demo.*.

The pattern *.TEdit specifies all files in the connected directory with the extension TEdit. It is equivalent to *.TEdit.*.

The pattern *.TEdit; specifies only the newest version of all files in the connected directory with the extension .TEdit.

The pattern <NoteCards>A*E specifies all files in directory NoteCards whose names begin with A and end with E and have any extension.

The pattern {DSK}<NoteCards>*MY* specifies all files in directory {DSK}<NoteCards> whose names contain the substring MY and any extension.

Using the FileBrowser Window

The FileBrowser window has six major subwindows, which from top to bottom are as follows.

Enumerating {DSK}/usr/users/kmount/NC/DOC/ENVOS/1.1/WORKING/*.*;* ...done			
{DSK}/usr/users/kmount/NC/DOC/ENVOS/1.1/WORKING/*.*;* at 12:26			
Total: 32		Deleted: 0	
Name		Created	
000-OUTLINE.TEDIT;13		4-Apr-89 11:28:15	
001-TITLE-PAGE.TEDIT;7		6-Mar-89 14:50:11	
002-PREFACE.TEDIT;15		17-Feb-89 09:52:39	
003-TOC.TEDIT;2		8-Mar-89 14:00:07	
01-INTRODUCTION.TEDIT;10		23-Feb-89 13:32:25	
02-SYSTEM-REQUIREMENTS.TEDIT;21		23-Feb-89 11:38:09	
03-SOFTWARE-INSTALLATION.TEDIT;22		23-Feb-89 13:40:36	
04-SYSTEM-USE-ISSUES.TEDIT;21		7-Mar-89 12:05:51	
05-NOTECARDS-BASICS.TEDIT;48		5-Apr-89 16:20:08	
06-BUILDING-NC-STRUCTURES.TEDIT;13		4-Apr-89 10:40:25	
Info Options			
Length	Pages	Created	Read
ByteSize	Type	Written	Author

Figure 14-1. FileBrowser Display

Prompt window

This topmost subwindow is where FileBrowser prints messages about what it is doing and receives input from you. Its contents are cleared before every command.

Tally window

This subwindow immediately below the prompt window keeps a running tally of the total number of files listed in the window and the number of files that you have marked for deletion. In addition, if one of the attributes you are displaying is a size attribute (Pages or Length, as in the INFO menu, described below), this window maintains a tally of the total number of pages in the files listed and the files marked for deletion.

This window also has a title bar across the top identifying the pattern you specified and the time at which the directory enumeration was performed.

The window is blank while the files are being enumerated.

Browser window

This is the principal subwindow, in which the files matching the specified pattern are listed (see Figure 14-1). Each file's name appears at the left, and various attributes of the file are displayed in columns to the right. A title bar across the top of the browser window identifies the contents of each column (e.g., Name, Pages, Created). The files are listed in alphabetical order, with multiple versions of the same file listed in decreasing version order; i.e., the newest version appears first. The width of the column listing the file names is initially chosen to be appropriate for average-sized file names. If the files you asked to browse have particularly long names, then when FileBrowser has finished listing all the files it may choose to redraw the browser window with the attribute columns moved farther to the right to accommodate the longer file names.

Command menu

This menu appears vertically along the right side of FileBrowser window (under a title bar "FB Commands") and lists the commands that you may select to perform

operations on the files in the browser, or to change the appearance of the browser. Most of the commands operate on the set of currently selected files (see the section *Selecting Files* below). Some commands have subcommands, as indicated by the small triangle alongside them, which can be selected by holding down the left mouse button and sliding the mouse to the right over the triangle.

Info menu

An additional subwindow, the Info menu, is not normally displayed. It is used to change the set of file information (attributes) displayed in the browser (see the *Getting Information About Files* section below).

Scroll bar

If there are more files in the listing than fit at one time in the browser window, you can scroll the browser window to view more files. Slide the mouse cursor out the left side of the browser window to get the scroll bar and press the left mouse button to scroll the region up and the right mouse button to scroll it down. Pressing a mouse button when the cursor is near the bottom of the scroll bar will scroll the region by larger increments than when the cursor is at the top.

You can also press the middle mouse button in the scroll bar to move the listing to the place that corresponds to that position in the scroll bar. For example, pressing the middle mouse button when the cursor is at the bottom of the scroll bar will display the end of the listing. This quick-scrolling technique is called *thumbing*. The gray box in the scroll region indicates where the currently displayed contents are, relative to the entire contents of the browser.

Similarly, if there is more attribute information than fits in the browser window, you can scroll the browser window horizontally to view the rest of the attribute information. To do this, slide the mouse out the bottom of the browser window to get the horizontal scroll bar. The left button scrolls to the left, the right button to the right.

Selecting Files

Most FileBrowser operations are performed by selecting a single file or set of files, then giving a command that specifies what you want to do with the selected files. The current selection is indicated by a small triangle in the left margin of the browser next to each selected file (see Figure 14-2).

```
| 003-TOC.TEDIT;2
| ▶01-INTRODUCTION.TEDIT;10
| 02-SYSTEM-REQUIREMENTS.TEDIT;21
| ▶03-SOFTWARE-INSTALLATION.TEDIT;22
| ▶04-SYSTEM-USE-ISSUES.TEDIT;21
| ▶05-NOTECARDS-BASICS.TEDIT;48
| 06-BUILDING-NC-STRUCTURES.TEDIT;13
```

Figure 14-2. Four Selected Files Marked by Triangles

To select one file, point to any part of the line (which lists the file name and its attributes) and press the left mouse button. If other files are already selected, this unselects them. A file selected with the left mouse button is always the only selection.

To add a single file to the current selection, press the middle mouse button at any place in the line. The file is selected without unselecting any other file.

To remove a single file from the current selection, hold down the CONTROL key and press the middle mouse button at any place in the line. The file is unselected without affecting any other file.

To extend the selection to include a group of contiguous files, that is, to select all the files between a file and the nearest already selected file, press the right mouse button on any part of the line. You can only extend the selection from the first selected file upward, or the last selected file downward. In addition, files marked for deletion are not normally selected when you extend.

If you want to include all files, both deleted and undeleted, hold down the CONTROL key while extending the selection.

Some lines in a FileBrowser display are directory-only lines. These lines are slightly indented and name the directory and subdirectory to which the files listed below that line belong. You cannot select in these lines, though you can copy-select them (see the Copy-Selecting Files section below).

Commands that Require Input

Some FileBrowser commands require input from you. For example, the **Copy** command requires that you supply a destination file name. When a command requires input, FileBrowser prints a prompt message in its prompt window. This is usually followed by a default answer. If you want the default answer, press RETURN to finish the input. If you want to specify a different answer, simply start typing it; the default answer is erased and your answer replaces it.

Alternatively, you can modify the default answer by backspacing over individual letters, or typing CONTROL-W to back up over complete words. Type CONTROL-Q to erase the entire answer. You can also use the mouse to edit your answer, using the same rules as followed by the Executive (see the documentation of TTYIN). Briefly, the left mouse button positions the caret at a character boundary; the middle mouse button positions the caret at the nearest word boundary; and the right mouse button deletes the characters between the caret and the mouse.

When you have finished, position the caret at the end of your answer, if it isn't there already, and press RETURN. You can also type CONTROL-X to finish your answer even if the caret isn't at the end.

If you change your mind and want to abort the command, supply an empty input; i.e., if there is an answer in progress, backspace over it or type CONTROL-Q to erase it, then press RETURN. FileBrowser prints "aborted" and aborts the command. In most situations, the CONTROL-E interrupt can also be used to abort your answer.

While you are typing an answer, you can copy-select file names out of the browser (or any other browser), as described below in the Copy-Selecting Files section. This can be useful, for example, if you wish to rename a file to a similar name in the same directory, or move a file into a subdirectory listed in the browser.

Aborting Commands

During commands of indefinite duration, such as **Recompute** or **Copy**, FileBrowser adds another command to the browser, **Abort** (see Figure 14-3).

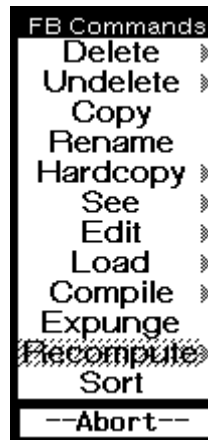


Figure 14-3. **FileBrowser** Menu with the Abort Option Attached

Clicking on the **Abort** command will immediately abort the current operation. Aborting some commands can take a little while, as FileBrowser may need to do some cleaning up, so the **Abort** command is greyed out during this time to show you that it is doing something.

Quitting the FileBrowser

To quit a FileBrowser, close the browser window. If any files have been deleted but not expunged, a small menu pops up listing two options: **Expunge deleted files** and **Don't Expunge** (see Figure 14-4). If you select **Expunge deleted files**, the files will be expunged before the window closes. If you select **Don't Expunge**, your deletions are ignored. If you click outside the menu, no action is taken, and the **Close** command is aborted.

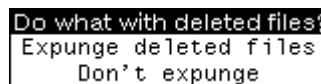


Figure 14-4. **Do what with deleted files?** Menu

If you have finished with FileBrowser only temporarily and want to put it aside to work on later, you can shrink the browser by selecting the **Shrink** command from the right-button background menu. The browser shrinks to an icon which displays the file pattern inside the browser (see Figure 14-5). If any files are marked for deletion, you are prompted with the same menu of expunge options as when you close a browser.



Figure 14-5. FileBrowser Shrink Icon

Copy-Selecting Files

You can copy-select file names from a FileBrowser into other windows, such as TEdit windows. Hold down the COPY or SHIFT key while selecting a name in the window. The full name of the file is inserted as if you had typed it where the input caret is flashing. You can also copy-select in a directory-only line, in which case the full directory name is inserted in your type-in.

Getting Hardcopy Directory Listings

You can get a hardcopy listing of the directory displayed in a FileBrowser by using the regular window **Hardcopy** command. Press the right button in FileBrowser's prompt window or tally window and select **Hardcopy** from the menu. FileBrowser produces a hardcopy listing of the files and the attributes displayed in the browser.

If the browser displays a large number of attributes, or your default printer font is too large, the listing may not accommodate all the attributes on one line, making the listing less readable. You may want to make the listing with fewer attributes, or use a smaller font for the listing.

FileBrowser Commands

Delete, Undelete

Removing a file from the file system using FileBrowser is a two-step process.

1. Mark the file or files for deletion.
2. Issue the **Expunge** command.

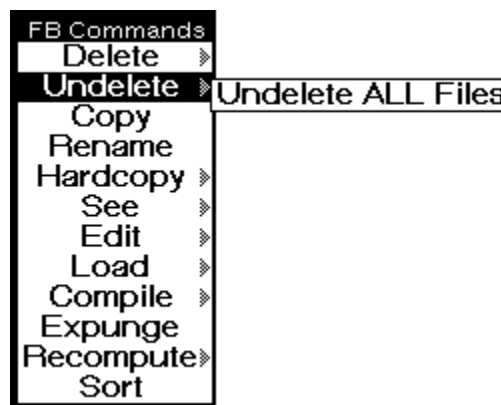
You can change your mind and undelete any of the files any time between the deletion and the expunge.

To mark a file or files for deletion, select them, then choose the **Delete** command. FileBrowser draws a line through the deleted files (see Figure 14-6). It also adjusts the numbers in the tally window to show how many files are marked deleted and how many pages they contain. This shows how much file space you will regain when you issue the **Expunge** command.

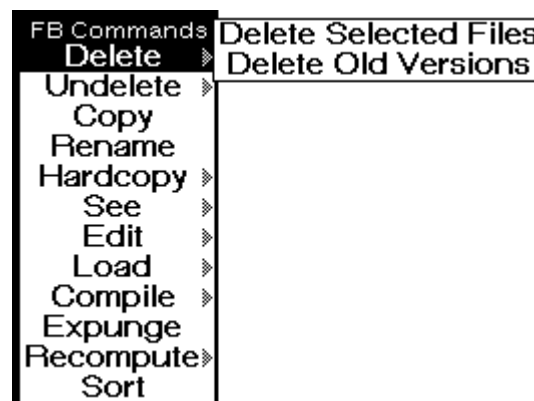
```
11-SYSTEM-CARDS.TEDIT;5
12-MENUBOX-ICON.TEDIT;11
13-PARAMETERS.TEDIT;9
14-FILEBROWSER.TEDIT;10
▶14-FILEBROWSER.TEDIT;9——
14-FILEBROWSER.TEDIT;8——
15-OTHER-TOOLS.TEDIT;27
16-PRINTING.TEDIT;52
```

Figure 14.6 Two Files Marked for Deletion

To undelete a file or files (i.e., to remove the deletion mark), select them, then choose the **Undelete** command. The lines through the files are removed, and the tally of deleted files is updated. The **Undelete** command has a single subcommand, **Undelete ALL Files**, which undeletes all the files in the browser, independently of whether they are selected (see Figure 14-7). This is useful if you completely change your mind about deleting any files.

Figure 14-7. **Undelete** Submenu

The **Delete** command has a useful subcommand, **Delete Old Versions**. When you have been editing a file in the text editor and performing repeated **Put** commands, multiple versions of the file accumulate, each more recent version denoted by a higher version number. The **Delete Old Versions** command is used to delete excess versions of the files displayed in the browser.

Figure 14-8. **Delete** Submenu

To use this command, press the left mouse button down on the **Delete** command and slide the cursor out to the right, selecting the **Delete Old Versions** command. Unlike the **Delete** command (or **Delete Selected Files**, the equivalent subcommand), the **Delete Old Versions** command operates on all the files in the browser. FileBrowser prompts for the number of versions of each file that you wish to retain. It offers the default of one version. You can accept the default or you can type a different number of your choosing, followed by a carriage return. FileBrowser then marks for deletion all but the most recent *n* versions of all the files in the browser, where *n* is the number you specified. Before issuing the **Expunge** command, you can, if you wish, scroll through the browser, undeleting any particular files for which you wish to retain more versions than you specified.

The **Delete Old Versions** command is sometimes useful even when you are not planning to actually expunge the files. This is because of the way extending the selection avoids deleted files (see the Selecting Files section above).

For example, if you wanted to copy only the most recent version of all the files in the browser to another location, you could do the following:

1. Use the **Delete Old Versions** command, retaining just one version. This marks deleted all files but the newest version of each.

2. Go to the start of the browser and select the first file, then scroll to the end of the browser and press the right mouse button to extend the selection to the end of the browser. You have selected exactly the newest version of each file.
3. Use the **Copy** command to copy those files.
4. Use the **Undelete ALL Files** command to undelete all the old versions.

Copy

Use the **Copy** command to copy an entire file or set of files to another file system location; for example, from your disk to a file server. Select the file(s) you wish to copy, then select the **Copy** command. FileBrowser prompts you to supply a destination.

If you selected just one file, FileBrowser prints the old name and offers a default, which consists of the same file name and either the same directory that was last used in a **Copy** or **Rename** in this FileBrowser, or the connected directory if this is the first use of **Copy** in this FileBrowser (see Figure 14-9). You can accept the default or supply your own destination file name. If you supply just a directory specification, e.g., *{SERVER}<DIRECTORY>*, the file is copied to that directory under its current name. If you supply a complete name, the file is copied to that exact name.

Copy file {DSK}/usr/users/kmount/NC/DOC/ENVOS/1.1/WORKING/14-FILEBROWSER.TEDIT;10 to new file name: {Eris}<NoteCards>Doc>ENVOS>1.1>WORKING>14-FILEBROWSER.TEDIT				
{DSK}/usr/users/kmount/NC/DOC/ENVOS/1.1/WORKING/*.;* at 13:18 Thu 6-Apr FB Commands				
Total: 34 / 4734 pages		Deleted: 2 / 652 pages		
Name	Pages	Created		
000-OUTLINE.TEDIT;13	31	4-Apr-89 11:28:15		
001-TITLE-PAGE.TEDIT;7	12	6-Mar-89 14:50:11		
002-PREFACE.TEDIT;15	24	17-Feb-89 09:52:39		
003-TOC.TEDIT;2	47	8-Mar-89 14:00:07		

Delete
Undelete
Copy
Rename
Hardcopy
See

Figure 14-9. Copy Command Confirming Name of File to which to Copy

Note: Unless you specify a version number in the destination file name, the version number of the new file will be 1, or one higher than the highest existing version of the file in the destination directory, independent of the version number of the old name.

Even files marked for deletion can be copied.

If you selected several files, FileBrowser notes how many files you wish to copy and offers as a default destination the connected directory. You can accept the default or supply a different directory. All the files are copied to that directory under the names they currently have.

You must supply a directory specification, e.g., *{SERVER}<YOURDIRECTORY>*, rather than a complete file name, since you cannot copy multiple files to the same name. If you mistakenly type a file name, rather than a directory specification, FileBrowser prints an error message and aborts the command.

If you want to copy files from different subdirectories, FileBrowser prints a message in its prompt window asking if you want to preserve the subdirectory structure at the destination. If you answer **Yes**, the names at the destination will include not just the root name of each source file, but also all the subdirectory names below the greatest subdirectory prefix common to all the selected files (this common prefix is displayed as part of the question). If you answer **No**, the names at the destination are formed solely from the root name of each file (the name displayed in the browser), ignoring any

directory information each name might have. This can cause multiple files with the same root name to be copied into the same destination name (but with different version numbers, of course).

When copying (or renaming) multiple versions of the same file, FileBrowser does the copying in order of increasing version number, so that the versions at the destination are in the same relative order as at the source.

As each file is copied, FileBrowser prints a message giving the full name of the new file. If a file with the chosen name already exists, the new file's version number will be one higher; otherwise it will be version 1 (one). The new file will have the same creation date as the original file. If the destination file happens to be one that matches the pattern of the files in the browser, the new file is inserted in the appropriate place in the browser display. However, if it matches the pattern of some other FileBrowser, it is not inserted in that other browser's display (in other words, FileBrowsers do not know about each other). You would have to **Recompute** the destination FileBrowser to see that the file was copied into it.

Rename

The **Rename** command is used for changing the name of a file or group of files, or for moving a file or group of files to a different directory.

The **Rename** command is used in exactly the same way as the **Copy** command. If you rename a single file, you can supply a complete new name or just a directory; if you rename several files, you must specify a directory. As each file is renamed, FileBrowser prints a message giving the file's new name and removes the file from the browser display. If the new name belongs in the same browser, it is inserted in the appropriate place. If a file could not be renamed, an appropriate message prints in the FileBrowser prompt window. The reasons for the failure of a renaming operation are roughly the same as for the failure of an **Expunge**; the file is open, or you do not have the access rights needed to rename the file.

Note: If the destination of the rename is on a different file system than the original file, changing its name is equivalent to copying the file to its new name and then deleting the original file.

Hardcopy

You can print text files, TEdit, Sketch, InterPress, and PostScript files from FileBrowser. Select the appropriate file or files, then select the **Hardcopy** command. The **Hardcopy** command determines what type of file you are printing and calls the appropriate function for printing that file. Then the files are printed one at a time on your default printer. The prompt window displays status messages telling you when files are being printed and when they are done (if your printer is one that provides this status service).

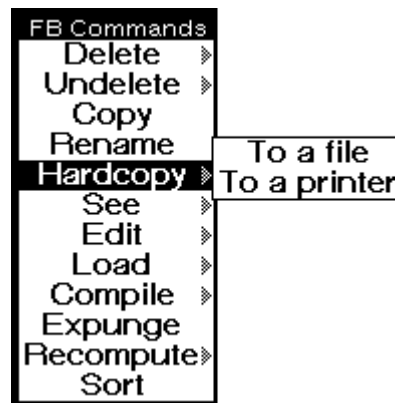


Figure 14-10. **Hardcopy** Submenu

You may specify printing to a file or to a printer other than the default printer by means of a submenu from the **Hardcopy** command (see Figure 14-10). This menu is the same as the one on the **Hardcopy** command in the background menu.

Selecting **To a printer** presents you with a choice of printers from a menu. Selecting **To a file** prompts you to supply a file name. If you selected a single file, you must specify a single hardcopy file name (or accept the offered default). If you selected multiple files, then you must specify a pattern with a single asterisk somewhere in the "name" field, for example, *.INTERPRESS or Hardcopy-*.IP. The output file names are constructed by merging the pattern with each selected file name. If the name includes an extension that implies the type of print format (e.g., .IP or .INTERPRESS implies the Interpress print format), then a file of the specified type is made automatically. Otherwise, you are prompted to supply a print format type.

Notes: For files stored on servers not supporting random access, FileBrowser is currently unable to determine that a file is in TEdit format unless the file has the extension .TEDIT. Therefore you should use TEdit to hardcopy TEdit files with other extensions. Use FileBrowser's **Edit** command (to call TEdit), then the **Hardcopy** command either from the TEdit Expanded Menu or from the right-button menu.

To obtain a hardcopy of the directory itself, use the **Hardcopy** command from the right-button window menu. See the Getting Hardcopy Directory Listings section.

See

When you browse a directory you sometimes want to see a file before printing or performing some other operation on it. To do this, select the file, then select the **See** command from the command menu. FileBrowser will prompt you to open a window by presenting you with a dashed rectangle and printing a message in the system prompt window. The window will be blank until FileBrowser starts printing the contents of the file in it.

There are actually four different **See** commands (see Figure 14-11). The **Fast SEE Pretty** and **Fast SEE Unformatted** commands let you quickly see the contents of a file, but not do anything fancy, such as scroll around at random in the file. The slower **Scrollable & Pretty** command does let you scroll, and if the file contains formatting information of a kind that FileBrowser knows about (via the editors you have loaded), you see the file formatted. However, this command does much more work, and may

take a bit longer to show you even the first line of the file. The **Filebrowse** command is for use on files that are directories; it is described in the next section.

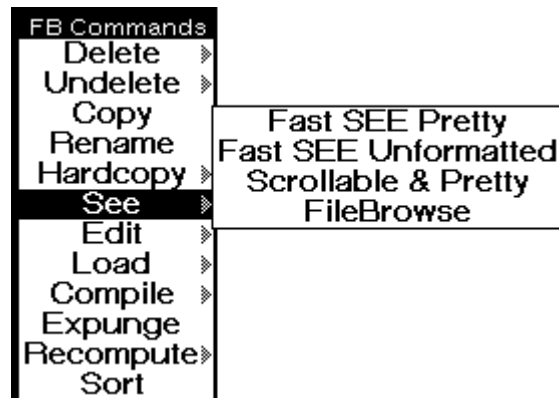


Figure 14-11. See Submenu

The **Fast SEE Pretty** and **Fast SEE Unformatted** commands display the selected file in the display window one windowful at a time. When the file fills the window, a small menu appears at the bottom-left corner of the window (or top-left if your display window is at the bottom of the screen) giving you the option of seeing more of the file or aborting the **See** command. If you issued the command with more than one file selected, you also have the choice of aborting just the display of this file or the entire **See** command (see Figure 14-12).

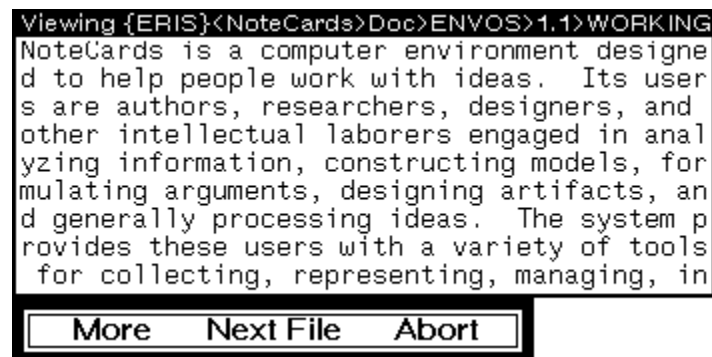


Figure 14-12. See Window Control Menu

If you select **More**, the **See** command displays another windowful of the file. If you select **Next File**, the **See** command closes this file and goes on to display the next file in the current selection. If you select **Abort**, the entire **See** command is aborted. You can also abort the **See** command by closing the display window.

The next time you give a **Fast See Pretty** or **Fast SEE Unformatted** command, the same window is reused.

The only difference between the **Fast See Pretty** and the **Fast SEE Unformatted** commands is the manner in which the characters of the file are processed as they are displayed.

The pretty (formatted) version interprets certain control characters found in source files as font change commands, and interprets certain multibyte sequences as representing

characters in the Xerox extended character set. It also squeezes out blank lines and shrinks the indentation of indented lines in order to better fit the text in a window that is generally much narrower than the standard file width. The formatted version is thus most appropriate for viewing source files and files containing plain text.

The unformatted version of the **See** command does no special processing on the characters. It simply displays each eight-bit byte as a single character, uninterpreted. This means bytes that do not represent normal printing characters may be displayed as black boxes, in the form ^x or #x, or as a flashing of the window (for the byte that represents the ASCII "bell" character). The unformatted version is most appropriate for viewing binary files that also contain text portions worth seeing; e.g., compiled files (those with extension .LCOM) or Interpress masters (extension .IP or .INTERPRESS).

The **Scrollable & Pretty** command views a file in a different way. This command brings up a new read-only TEdit window for viewing a file (only if TEdit is loaded in your system; otherwise, **Scrollable & Pretty** reverts to fast format). You can scroll and copy-select the file's contents at will, as with any TEdit window. If the file is a NoteCards source file, its contents are first formatted into a TEdit document, so that all the font information is retained. This formatting, however, can take a long time for a large file. For other kinds of files, the **Scrollable & Pretty** command is exactly like viewing the file in a regular TEdit window, except that you can't edit it. If you want to edit a file, use the Edit command instead of the See command.

You can keep the display window used by the **Scrollable & Pretty** command open as long as you like. The command uses a different window for each file you select. Simply close the window with the standard right-button window menu when you are finished with it.

Filebrowse

The **Filebrowse** command is a subcommand of **See** used to view a subdirectory in its own FileBrowser window. The selected file must be a (sub)directory. Subdirectory files appear in browsers on XNS file servers when the depth is finite (see the **Set Depth** command), and their names always end in ">". On Sun servers subdirectory file names end in "/" (see Figure 14-13).

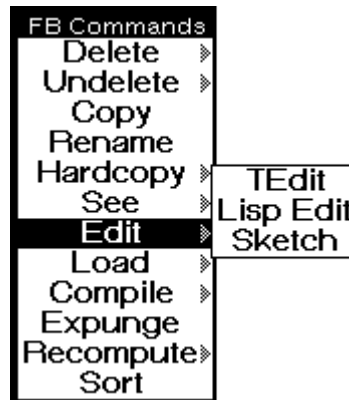
```
| medley/  
| ►nc/  
| NOTES/  
| PROGS/  
| setup/  
| sysouts/
```

Figure 14-13. Sample Listing of Subdirectories on a Sun server
(indicated by trailing slashes)

The **Filebrowse** command prompts for a region for a new FileBrowser window group. It then enumerates the contents of the selected subdirectory, to the same depth as the main browser used, if any.

Edit

The **Edit** command invokes an editor on the selected file. To specify an editor explicitly, use one of the commands on the submenu (see Figure 14-14).

Figure 14-14. **Edit** Submenu

To start up a TEdit editor on a selected text file, select **Edit** with the left mouse button. If you have recently closed a TEdit window, then TEdit will probably reuse that window; otherwise, you will be prompted to create an editor window. TEdit only remembers the most recently abandoned window, however, so if you issue the **Edit** command when you have several files selected, you will be prompted to create windows for all but the first file.

The subcommand **Lisp Edit** is for people extending the NoteCards environment. Do not use this command unless you are extending the NoteCards system.

If you select the main **Edit** command, without sliding off to the submenu, then FileBrowser's default editor, TEdit, is called.

Load

The **Load** command is for people extending the NoteCards system. Do not use this command unless you are extending the NoteCards system.

Compile

The **Compile** command is for people extending the NoteCards system. Do not use this command unless you are extending the NoteCards system.

Expunge

If you are sure you want to delete files permanently, choose the **Expunge** command. The **Expunge** command is grayed while FileBrowser expunges the files that were marked for deletion by the **Delete** command. As each file is removed from the system, it is removed as well from the browser display, and the tally of total number of files and number of deleted files is updated, so you can see the progress of the command.

If for some reason a file can not be expunged, FileBrowser prints a message saying so in its prompt window, but continues to expunge the other files. The main reasons that prevent a file from being expunged are its being opened, either by you or some other user, or your not having the access rights required to delete it (if it is on a file server). See the Troubleshooting Problems with FileBrowser section below.

Note: The **Expunge** command is not affected by the current selection; it operates only on files marked for deletion, whether currently selected or not.

Recompute

FileBrowser's display shows those files that existed and matched the specified pattern at the time you created the browser. If you want the browser to reflect the latest state of the file system, use the **Recompute** command.

For example, if you open a FileBrowser on your directory, then save several versions of a TEdit file on that directory, the file listing will not display the new versions until you **Recompute**.

The **Recompute** command operates exactly as when you started up FileBrowser initially: it clears the display and tally windows, then enumerates the files matching the pattern. The **Recompute** command in the menu is grayed until the enumeration is finished. During this time you cannot scroll or perform any other operations on the browser. However, you can close the window if you want to abort the command and throw away the browser.

If any files are marked for deletion at the time you request a **Recompute**, FileBrowser will present the choice of expunging or undeleting the files, just as it does when you want to quit the browser (see the Quitting the FileBrowser section above).

The **Recompute** command also has a menu of subcommands that allow you to list different files or different information for the same set of files (see Figure 14-15).

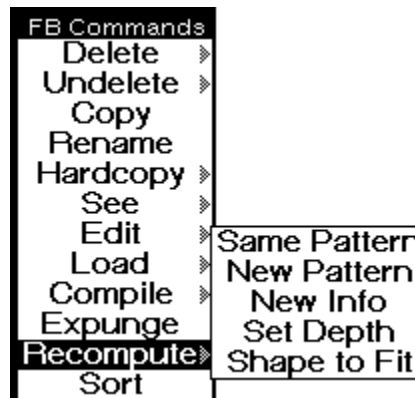


Figure 14-15. **Recompute** Submenu

Same Pattern	Is the same as the main Recompute command, i.e., it enumerates the files matching the same pattern as before.
New Pattern	Lets you change the pattern, i.e., browse a new set of files. FileBrowser prompts you to supply a new file name pattern and offers the old pattern as an initial default. You can either type an entirely new pattern, replacing the one offered, or delete the old pattern one character at a time by backspacing. Press the carriage return when you have finished specifying the pattern. FileBrowser then enumerates the files matching this pattern, just as with the Recompute command. You can abort the command with the Abort button, or by erasing the whole pattern (by backspacing or using CONTROL-Q) and then pressing the RETURN.

- New Info** Lets you change which attributes the browser displays. It is described in the next section.
- Set Depth** Lets you change the depth to which FileBrowser enumerates a directory on an XNS file server. It is described in the section Set Depth.
- Shape to Fit** Reshapes the FileBrowser window so that all the attributes in the display are visible at once, eliminating the need to horizontally scroll the window to get at all the information.

New Info

FileBrowser displays some file attributes, or information about the file, next to each file in the browser display. Ordinarily, the attributes displayed are the size of the file in pages, its creation date, and its author. You can change the attributes displayed in a particular browser window by using the **New Info** command.

To use the **New Info** command, select it from the submenu of the **Recompute** command. FileBrowser opens up an additional subwindow, the "Info Options" window, below the display window. This subwindow contains a menu of attributes, with the current defaults shaded. Selecting a shaded item unshades it; selecting an unshaded item shades it. When you have selected all the attributes you wish to see displayed, issue either the **Recompute** or the **New Pattern** command. The files will be listed with the new information you requested. The Info Options window stays open—you can close it at any time.

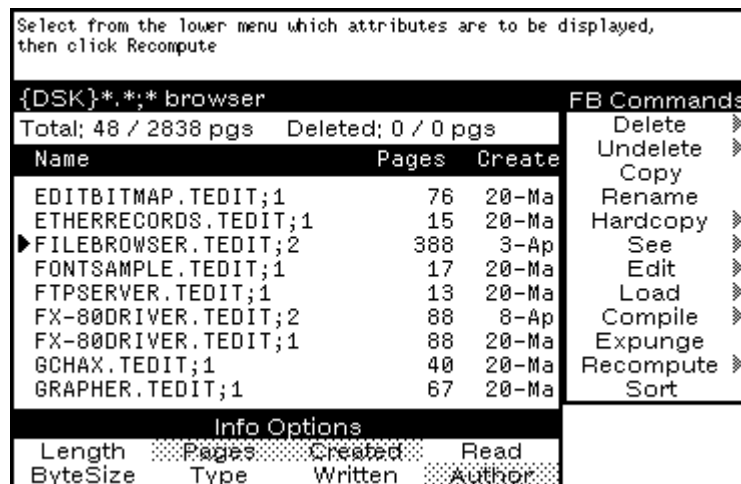


Figure 14-16. FileBrowser with Info Options Window

The Info Options items have the following meanings:

- Created** The date and time that the content of the file was created. This date changes whenever the file is modified, but does not change when a file is copied or renamed.
- Written** The date and time the file was last written to the file system. This date is never older than the Created date, but it can be newer if the file is copied, unmodified, from one file system to another.

Read	The date and time the file was last read. This attribute may be blank if the file has never been read.
Author	The login name of the person who wrote the file, or last modified it.
Length	The length of the file in (usually eight-bit) bytes.
Pages	The number of 512-byte pages in the file. On some servers, this attribute is blank if the file is empty.
ByteSize	The size (in bits) of the bytes in the file.
Type	A value indicating what kind of data the file contains. The usual values of this attribute are <code>TEXT</code> , meaning the file contains just characters, or <code>BINARY</code> , meaning the file contains arbitrary data. Some servers have additional types, such as <code>INTERPRESS</code> for files in Interpress format.

Set Depth

XNS file servers support a feature that allows enumerating a directory to a user-specifiable depth. The "depth" of a file reflects the number of subdirectories between it and the root of the enumeration, i.e., the directory or subdirectory you gave in the pattern to FileBrowser, not counting any containing wildcards (asterisks). The immediate descendants of the root are at depth 1, files in subdirectories of depth 1 are at depth 2, and so on.

Ordinarily, FileBrowser enumerates a directory to the default depth, which is usually unlimited. To enumerate a directory to a different default, use the `FB` command with argument `:DEPTH n`, for some positive integer *n*, or `T` for unlimited depth. To change the depth in an existing FileBrowser, use the **Set Depth** command, a subcommand under the **Recompute** command. The command offers you a menu of choices:



Figure 14-17. **Set Depth** Menu

Global default means use the default depth, overriding the depth at which this browser was last enumerated. **Infinite** means use no depth limit (same as depth `T`). **1** and **2** are common depth choices; to choose some other numeric value, select **Other** and enter the value via the displayed keypad.

The **Set Depth** command does not affect the current display. It takes effect the next time you use the **Recompute** or **FileBrowse** commands from the same browser.

During a **Recompute**, if a subdirectory appears at the specified maximum depth, its descendants are not enumerated; rather, the subdirectory itself appears as an entry in the browser display. This entry can be selected, just like a file, but only a small number of commands can be used on it: you can **Rename** it, you can **Delete** it if it has no descendants, and you can **FileBrowse** it. It has attributes, just as ordinary files do. Its page size is the size of the entire subtree rooted at the subdirectory.

Note: Depth currently affects only XNS servers; all other devices ignore it and enumerate to their own default depths. In addition, due to a bug in XNS Services 10, depth is ignored for nontrivial patterns, i.e., anything but "**.*".

Sort

The **Sort** command allows you to sort the files in the browser by any attribute of the files displayed in the browser. Selecting **Sort** brings up a menu of attributes by which to sort. This menu includes all the attributes currently displayed in the browser (such as Creation Date, Author), plus the choice Name. For some attributes you can sort forward or backwards; the choice is on a submenu, and the default is generally in the order of numerically greatest (e.g., size) or most recent (e.g., creation date) first.

If the attribute you select is not Name, then the file names displayed in the browser will be reformatted to include their directory portion (if there are any subdirectories below the browser's main pattern), as the subdirectory information is no longer implicit in a file's position in the browser.

The sort order Name, Decreasing Version is the default order in which browsers initially are created.

Troubleshooting Problems with FileBrowser

When FileBrowser returns the message: No files in group *FILENAMEPATTERN* when you know those files exist, the file server is probably down or rejecting connections. If this is so, your only option is to wait until the server is functioning again, and then give the **Recompute** command. In the case of an Xerox file server, the enumeration of files can also fail if you do not have sufficient access privileges; this condition is usually noted by a message in the system prompt window.

When you try to expunge a file and FileBrowser displays the message: Can't expunge *FILENAME*, it may be because you don't have write access to the file, or someone else is reading the file. However, the most common reason is that the file is still open. Be sure to close any TEdit windows in which you may still be viewing the file. If you have recently issued a **Hardcopy** command for the file, a background process may still be working on the file.

[This page intentionally left blank]

15. OTHER TOOLS

Digital Clock

The digital clock allows you to keep track of the time in multiple time zones. The clock updates itself once a minute. Clicking the left mouse button inside any of the clock windows will cause the digital clock to update itself.

17 Feb 89 11:39am (Fri)		
Washington DC:	17 Feb 89	2:39pm (Fri)
Tokyo, Japan:	18 Feb 89	5:39am (Sat)

Figure 15-1. Digital Clock with Two Auxilliary Regional Time-zone Windows

Starting and Stopping the Digital Clock

To put a digital clock on the screen, bring up the background menu by holding down the right mouse button, position the mouse over the **Clocks** option and slide off to the right to bring up the Clocks submenu. Slide the mouse cursor over the **Digital Clock** choice and release the right mouse button (see Figure 15-2).



Figure 15-2. **Digital Clock** Option from the **Clocks** Submenu

Selecting the digital clock option will delete any existing digital clock and start a new one.

When the clock comes up, it will prompt you for an auxilliary time zone. If you click outside the "Enter Time Zone" menu, the clock will come up with just the local time zone.

ENTER TIME ZONE
Nome, Alaska
Honolulu, Hawaii
Marquesas Islands
San Francisco, California
Denver, Colorado
Houston, Texas
Washington DC
Buenos Aires, Argentina
Brasilia, Brasil
Rio de Janeiro, Brasil
Reykjavik, Iceland
Greenwich, England
Paris, France
Athens, Greece
Moscow, USSR
Riyadh, Arabia
Kabul, Afganistan
Kathmandu, Nepal
Bangkok, Thailand
Hong Kong
Seoul, South Korea
Tokyo, Japan
Sydney Austrailia
Aukland, New Zealand

Figure 15-3. Enter Time Zone Menu

To remove the clock from your screen and kill the clock process, simply close the clock window using the clock window's right mouse-button **Close** option.

Changing the Digital Clock

You can specialize the digital clock to suit your needs. You modify the digital clock by holding the middle mouse button down in the clock window. Holding this mouse button down in the main clock window brings up the menu shown in Figure 15-4. Holding the middle mouse button down in one of the regional time-zone windows brings up the menu shown in Figure 15-5.



Figure 15-4. Main-Window Middle-Button Menu

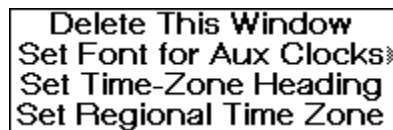


Figure 15-5. Auxilliary-Window Middle-Button Menu

Set Font

Brings up a series of menus which allows you to change the font in the main clock window.



Figure 15-6. Font Menu

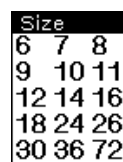


Figure 15-7. Font Size Menu



Figure 15-8. Font face Menu

Select each option you want in sequence. If you want to keep any particular aspect of a font, simply click outside that particular menu. Clicking outside all the menus, leaves the font exactly the way it was to begin with.

While the system is searching for a font in the font files, it displays the message, "Fetching Font."

Not all possible combinations of every font exist. When the clock cannot find a particular font, it displays the message "Font Not Found."

Set Time

Brings up a menu which allows you to set the time.

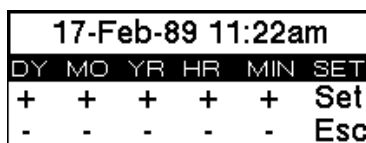


Figure 15-9. Set-time Menu

Each click on a + advances the counter for that category; each click on a - decreases the counter for that category. Selecting the **Set** option sets the time and closes the menu. Selecting **Esc** (Escape) closes the menu without setting the time.

Set Alarm

Brings up the set-time menu shown in Figure 15-9. The menu functions the same way it does for the **Set Time** option. After you have set the time, the clock will ask you for a message to associate with the alarm time. Type in your message then press RETURN.



Figure 15-10. Digital Clock with Alarm Time Message in Auxilliary Window

Quiet Alarm/Loud Alarm

Toggles the clock back and forth between, a loud/auditory alarm and a quiet/visual alarm.



Figure 15-11. Main Window Middle-Button Menu Showing **Loud Alarm** Option instead of **Quite Alarm**

The loud alarm causes the monitor to beep once a minute when the alarm is ringing. The quiet alarm causes the screen to flash once a minute when the alarm is ringing.

Delete Alarm Setting

Turns the alarm off when it is ringing. This option, shown in Figure 15-11, only appears when the alarm is set.

Shape to Fit

Resizes the window to the minimum size necessary for the font you are using.

The window should resize automatically whenever you change the font. If ever the window and font size get out of synch, you can force the window to resize itself by selecting this option.

12-Hour Clock/24-Hour Clock

Toggles the digital clock back and forth between 12- and 24-hour modes. Compare Figures 15-4 and 15-11 to see how these options appear in the menu.

Set Local Time Zone

Allows you to change the time zone shown in the main digital clock window. Selecting this option brings up the "Enter Time Zone" menu, shown in Figure 15-3, from which you can choose the appropriate time zone. Clicking outside this menu leaves the time zone unchanged.

Add New Regional Time Zone

Allows you to add an additional regional time zone to the digital clock display. Selecting this option opens a new auxilliary window on the bottom of the clock and then pops up the "Enter Time Zone" menu, shown in Figure 15-3, for you to select a time zone for the auxilliary window.

Delete This Window

Deletes the auxilliary window the mouse cursor was in when you pressed the middle mouse button to bring up the auxilliary-window menu.

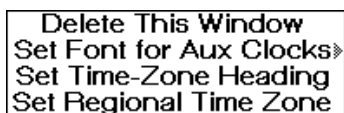


Figure 15-12. Auxilliary-Window Menu

Set Font for Aux Clocks

Changes the font for all the auxilliary clocks indepent of the main clock. Selecting the option brings up the series of menus shown in Figures 15-6, 15-7, and 15-8. Clicking outside of any menu leaves that aspect of the font the way it was originally.

Set Aux Clock Font In Just This Window

Changes the font for the auxilliary window the mouse cursor was in when you pressed the middle mouse button to bring up the auxilliary-window meu. Selecting the option brings up the series of menus shown in Figures 15-6, 15-7, and 15-8. Clicking outside of any menu leaves that aspect of the font the way it was originally.



Figure 15-13. Submenu for Setting Font in One Auxilliary Window Only

Set Time-Zone Heading

Prompts you for a new heading for the auxilliary window you were in when you pressed the middle mouse button to bring up the auxilliary-window meu. Type in the name followed by a carriage return.

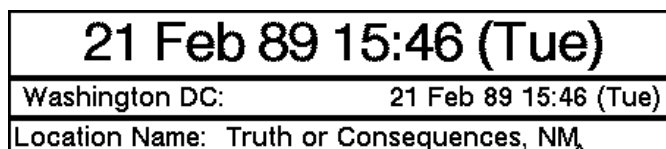


Figure 15-14. An auxilliary window prompting for a new heading.

Occasionally, the window isn't wide enough to display the whole name. In this case choose the option **Shape to Fit** from the main window middle-button menu.

Set Regional Time Zone

Brings up the "Enter Time Zone" menu shown in Figure 15-3, which allows you to change the time zone region. **Set Regional Time Zone** does not allow other than hour increments.

Analog Clock

The **Analog Clock** suboption off the **Clock** option on the background menu sets up an analog clock on your screen. The clock is updated once a minute.

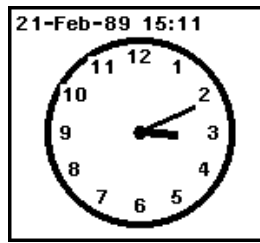


Figure 15-15 Analog Clock

Starting the Analog Clock

To put an analog clock on the screen, bring up the background menu by holding down the right mouse button, position the mouse over the **Clocks** option and slide off to the right to bring up the Clock submenu. Slide the mouse cursor over the **Analog Clock** choice and release the right mouse button.



Figure 15-16. Analog Clock Option from the Clock Submenu

The first time you bring the analog clock up in a sysout, it will prompt you to sweep out a window. Once you have done this this clock will always appear in its last size. If you want to resize the analog clock, choose the **Shape** option from the right button window menu, and sweep out the new shape for the clock.

Stopping the Analog Clock

To remove the clock from your screen and kill the clock process, simply close the clock window using the clock window's right mouse-button **Close** option.

Changing the Analog Clock

Holding the middle mouse button down inside the analog clock pops up the following menu.

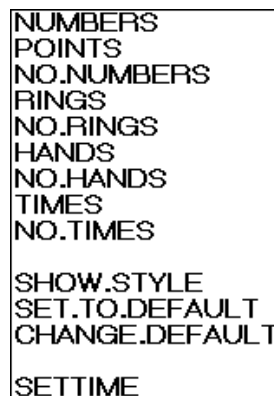


Figure 15-17. Analog Clock Menu

There are four independent properties which the user may control: the hands of the clock, time digits printed where the hands end, rings on the clock face, and 12 numbers around the outside of the clock face.

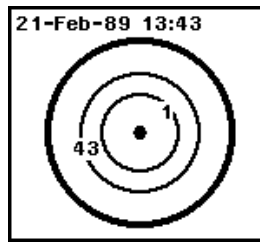


Figure 15-18. Analog Clock with No Points or Numbers, or Hands

NUMBERS

Removes points and adds numbers to the clock face.

POINTS

Removes numbers and adds points to the clock face.

NO.NUMBERS

Removes both numbers and points from the clock face.

RINGS

Adds rings to the clock face.

NO.RINGS

Removes rings from the clock face.

HANDS

Adds hands to the clock face.

NO.HANDS

Removes hands from the clock face.

TIMES

Adds time digits to the clock face.

NO.TIMES

Removes time digits from the clock face.

SHOW.STYLE

Prints the current style to the system prompt window.

SET.TO.DEFAULT

Resets the clock face to the default settings.

CHANGE.DEFAULT

Sets the default to the current style.

SETTIME

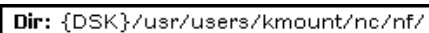
Forces the clock to go to the machine to ask it the time.

Known Problems

The clock will occasionally not respond to style changes. To force it to change, close the clock window and restart the clock from the background menu.

Directory Connector

The directory connector allows you to keep track of your currently connected directory and allows you to change the directory. It updates itself about once every ten seconds.



Dir: {DSK}/usr/users/kmount/nc/nf/

Figure 15-19. Directory Connector Window

Starting the Directory Connector

To put a directory connector on the screen, bring up the background menu by holding down the right mouse button, position the mouse over the **Directory Connector** option and release the right mouse button.

Stopping the Directory Connector

To remove the directory connector from your screen and kill its process, simply close the directory connector window using the window's right mouse-button **Close** option.

Changing the Directory Connector Fonts

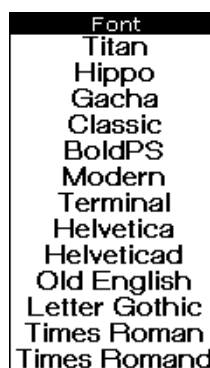
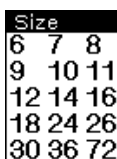
To change the directory connector font, bring up the background menu by holding down the right mouse button, position the mouse over the **Directory Connector** option, slide off the menu to the right, position the mouse over the **Change Font** option, and release the right mouse button.



Directory Connector » **Change Font**

Figure 15-20. **Directory Connector** option on the background menu showing the **Change Font** submenu option.

When you select the Change Font option you will be presented with a series of two menus asking you for a font and a size.

Figure 15-21. **Font** MenuFigure 15-22. **Font Size** Menu

While the directory connector is looking for a font, it prints the message "Fetching Font..." to the system prompt window.

Not all possible combinations of fonts are possible. If a font is not found the directory connector prints the message "Font Not Found." in the system prompt window.

Using the Directory Connector

The directory connector can be used in two ways.

Left Mouse Button

Clicking the left mouse button in the directory connector window causes the window to update itself.

You can shift-select the connected directory name out of the directory connector window with the left mouse button. You do this by pointing to the position in the window where you want the name of the directory to appear and clicking the left mouse button to position the caret cursor there, a FileBrowser prompt window for instance. Next, hold a SHIFT key down and click the left mouse button in the directory connector window. The name of the directory will appear where you positioned the caret cursor.

Middle Mouse Button

Clicking the middle mouse button in the directory connector window brings up a menu of directories to connect to.

Figure 15-23. **New Directory?** Menu

If the directory you want to connect to is on this list, click the left mouse button on it to connect to it.

If you change your mind and want to abort the operation, click outside the "New Directory?" menu.

If the directory you want to connect to is not on this list, click the left mouse button on *
* Connect to Other Directory * * and type the directory name into the prompt window
which will appear.



Figure 15-24. New Directory? Prompt Window

The prompt window will come up with your currently connected directory. The moment you start typing the system will erase the window and show what you are typing.

If you want to modify the name of your currently connected directory, backspace over as much of the name as you need to and type the name of the new subdirectory. Be sure to end it with a slash.

If you type in an invalid directory name, directory connector will beep and print the message "Not a valid directory name." to the system prompt window.

[This page intentionally left blank]

16. PRINTING

NoteCards includes facilities for printing in the PostScript language by Adobe, and the InterPress language by Xerox.

This chapter explains how to:

- Set up your printers
- Specify where your font directories are
- Deal with some common printer problems
- Customize your PostScript printer

Setting Up Printers

You can set up printers from either the background menu or the initialization file. For how to set up your printers with the initialization file, see Appendix C.

Adding Default Printers

To set up a new default printer from the background menu, press and hold the right mouse button in the background, place the mouse cursor on **Set Default Printer**, then slide off to the right and select **Add Default Printer**.

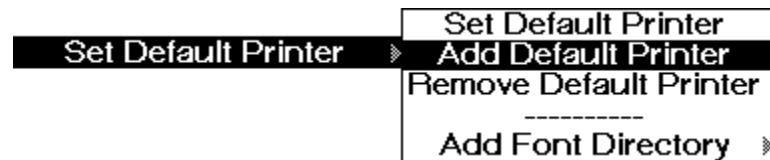


Figure 16-1. **Set Default Printer** Background Menu Option with its Submenu

A prompt window appears showing you the name of the current default printer and asking for the name of a new default printer.

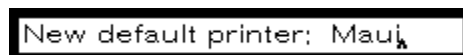


Figure 16-2. PostScript Printer Name Entered as New Default Printer

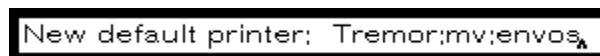


Figure 16-3. InterPress Printer Name Entered as New Default Printer

Enter the name of the new default printer. The moment you type the first character of the new printer name, the old name is deleted and replaced with what you type. If you don't want to make any changes, press RETURN. If you want to modify the name that is there, you can backspace over the name to erase part of it. If you press RETURN without entering a printer name, this operation is aborted.

When you press RETURN to end the name, a menu pops up requesting the printer type (Figure 16-4). Select one of the options with the left mouse button.



Figure 16-4. System Prompting for Printer Type

If you do not select any option (by clicking outside the menu) and the printer's type was previously defined, it remains the same. If the printer's type was not previously defined, and you do not select any option, the printer's type is left undefined, which will cause problems when you try to print. If you leave the printer type undefined the system will warn you (Figure 16-5).

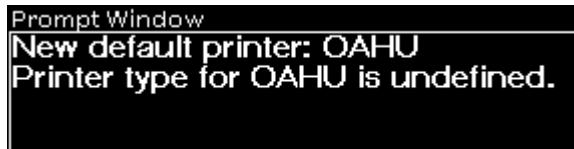


Figure 16-5. System Prompt Window with New Printer Status and warning Printer Type for Oahu is Undefined



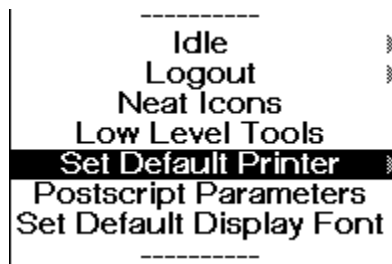
Figure 16-6. System Prompt Window with New Printer Status

At this point, if you have already specified the font directories, you should be able to print to any default printer.

Setting the Default Printer

The system maintains a list of printers, one of which is the default printer—the printer that files are sent to when no printer is explicitly specified.

To set the default printer, select the **Set Default Printer** option from the background menu (Figure 16-7).

Figure 16-7. **Set Default Printer** Option on the Background Menu

This causes the system to open a menu listing all the known printers (Figure 16-8).

Figure 16-8. **Choose Default Printer** Menu

Selecting one of these options will make that printer the default printer. The current default printer is the topmost option.

Once you have selected a new default printer, the system confirms your choice by printing the printer name and type to the system prompt window (Figure 16-6).

If the printer type is undefined, the system beeps and issues a warning (Figure 16-5).

Removing a Default Printer

If you make a mistake specifying a printer or want to remove a printer for some reason, you can use the **Remove Default Printer** submenu option on the background menu.

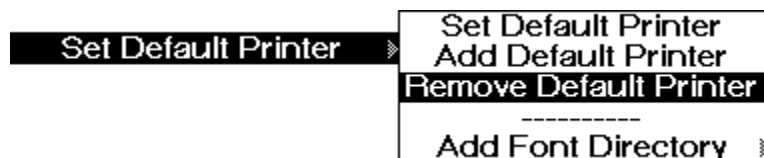


Figure 16-9. **Remove Default Printer** Submenu Option

Selecting this option brings up a menu of known printers. Selecting one of these deletes it from the known-printers list and makes the topmost remaining printer the default printer.



Figure 16-10. **Choose Printer to Remove** Menu

Setting Font Directories

As with printers, it is possible to set your font directories from either the background menu or your initialization file. We recommend that you set your directories from the initialization file. There are two reasons for this. Setting the font directories involves a lot of typing, and these directories are set correctly for you by the standard initialization file unless you have moved the font files. See Appendix C for how to set the font directories from the initialization file.

Adding a New Font Directory

To add a font directory from the background menu, press and hold the right mouse button in the background to bring up the background menu. Position the mouse cursor on the **Set Default Printer** option and slide off to bring up the submenu containing **Add Font Directory**.



Figure 16-11. **Add Font Directory** Submenu Options

Selecting **Add Font Directory** brings up a menu asking you to specify the font group you want to add a search directory to.



Figure 16-12. **Add to which directory group?** Menu

The NoteCards system supports one display and three hard copy modes. For each of these there is a specialized set of fonts. The fonts for the screen display are stored in the display font directories, the fonts for PostScript printers are stored in the PostScript font directories, those for InterPress printers are stored in the InterPress font directories, etc. When you add a new directory to one of these groups, you are providing the system with another place to search for fonts. For example, the display font directories typically include the following directories:

```
{DSK}/usr/local/lde/fonts/display/presentation/
{DSK}/usr/local/lde/fonts/display/presentation/
{DSK}/usr/local/lde/fonts/display/publishing/
{DSK}/usr/local/lde/fonts/display/printwheel/
{DSK}/usr/local/lde/fonts/display/miscellaneous/
```

When the system is looking for a new display font, it searches through each of these directories in sequence for the font. The fonts are broken up into separate directories for convenience and efficiency reasons.

Once you have specified the font directory group you want to work with, the system prompts you for the name of a new directory to search for fonts of that type.

A screenshot of a prompt box with a black border. Inside, the text reads "Add font directory: {DSK}/usr/local/lde/fonts/interpress/jis1/▲".

Figure 16-13. System Prompting for New Font Directory to Search for InterPress Fonts

Enter the name of the new font directory. The moment you type the first character of the font directory name, the old name will be deleted and replaced with what you type. If you don't want to make any changes, strike the carriage return key. If you want to modify the name that is there, you can backspace over the name to erase part of it. If you type a carriage return without a printer name, this operation is aborted.

Note : For Sun machines, the directory name always starts with {DSK}; subdirectories in the path name are separated by slashes; and the directory path name must end in a slash (/).

Showing the Font Directories

To view a particular set of font directories, use the **Show Font Directories** submenu option (Figure 16-14).

Like **Add Font Directory**, this first brings up a menu asking you for the directory group you want to show.

Figure 16-14. **Show which directory group?** Menu

Once you have selected the directory group, the system pops up a window which displays the font directories in that group. You can close this window using the standard window-menu close option or the window will close of its own accord after about two minutes.



Figure 16-15. Window Showing List of Display Font Directories

Removing a Font Directory

If you make a mistake entering a font directory name or for some other reason want to remove a font directory, use the **Remove Font Directory** option shown in Figure 16-13.

Like the **Add Font Directory** option, this first brings up menu asking you for the directory group you want to modify (Figure 16-16).

Figure 16-16. **Remove from which directory group?** Menu

Once you have selected the directory group, the system pops up a menu of all the directories currently in that group (Figure 16-17).

Figure 16-17. **Remove which directory?** Menu

Select one of these options to delete that directory. Click outside the menu to abort this operation.

Troubleshooting Common Printer Problems

This section discusses some of the more common breaks which occur when your printers are not properly set up.

Can't determine IMAGETYPE for {LPT}

This break occurs when you do not have any printers specified on the default printer list, or if you have not specified the printer type.



Figure 16-18. Break Window indicating System Cannot Find a Valid Printer

Type an up-arrow to exit the break and make sure that the printer and its type are properly set.

You can get into this situation by choosing the hardcopy **To a printer** suboption off the standard window menu and then selecting the **Other...** option from the **Which Printer?** menu.

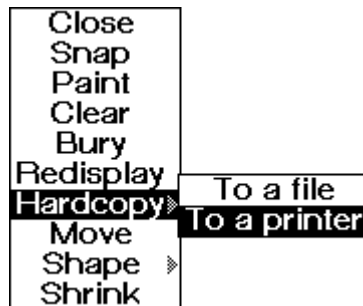


Figure 16-19. Standard Right Button Window Menu with Hardcopy **To a printer** Submenu Option Selected

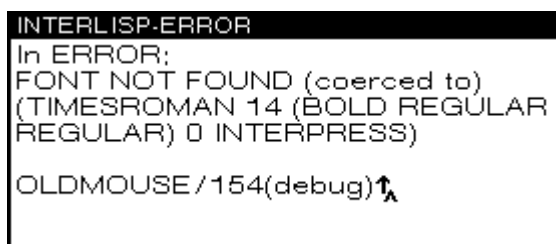


Figure 16-20. **Which printer?** Menu showing **Other...** Option Selected

The **Other...** option does not automatically prompt you for a printer type. Therefore, we recommend that you always add new printers using the background menu **Add Default Printer** submenu option.

Font not Found

This break occurs when you do not have one of your font directories set properly.



```

INTERLISP-ERROR
In ERROR:
FONT NOT FOUND (coerced to)
(TIMESROMAN 14 (BOLD REGULAR
REGULAR) 0 INTERPRESS)

OLDMOUSE/154(debug)↑
  
```

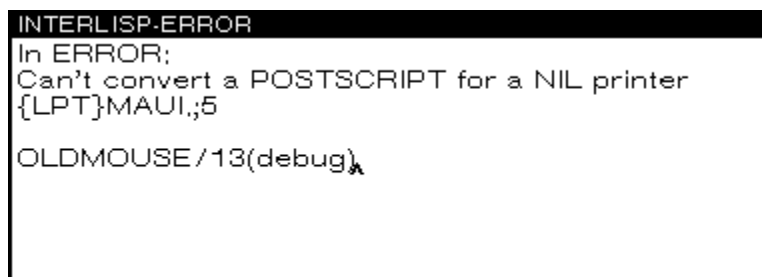
Figure 16-21. A break window indicating that the system cannot find the appropriate font.

In the example shown in Figure 16-21, the system was unable to find an InterPress font. Note that the font description contains the word InterPress. This implies that either there are no InterPress font directories set or that the given InterPress font directories do not contain that font.

Type an up-arrow to exit the break and make sure that the appropriate font directory is properly set. You can do this by selecting the **Show Font Directories** suboption off the **Set Default Printer** option on the background menu. If directories are not correctly set use the **Add Font Directory** and **Remove Font Directory** options to correct the errors.

Can't Convert a POSTSCRIPT for a NIL Printer

This break can occur if you have incorrectly typed the case of the printer name.



```

INTERLISP-ERROR
In ERROR:
Can't convert a POSTSCRIPT for a NIL printer
{LPT}MAUI,;5

OLDMOUSE/13(debug)↑
  
```

Figure 16-22. Break Window Showing System Cannot Find Appropriate Printer

Type an up-arrow to exit the break and, if you have entered a printer name in lowercase, try entering the same name in uppercase by selecting the **Add Default Printer** option off the **Set Default Printer** submenu on the background menu.

Customizing Your PostScript Printer

The PostScript driver allows you to set a number of parameters to modify the output to suit your needs.

Portrait vs. Landscape

There are two terms used to describe how an image is printed on a piece of paper which you need to know.

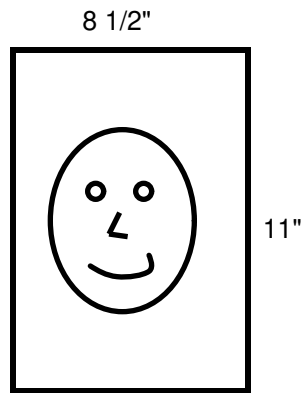


Figure 16-23. Portrait describes the way we normally hold a piece of paper

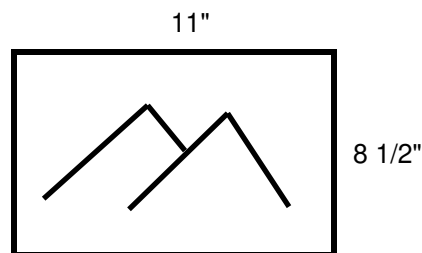


Figure 16-24. Landscape describes a piece of paper held on its side

Maximum Wild Font Size

Indicates the maximum point size that the system will indicate as available when looking for fonts.

The default value is 72.

Font Association Menu

Maps display font names onto PostScript font names. To change the mapping, click the left mouse button on a display font name in the left-hand column, then select a PostScript font from the a menu of PostScript fonts names which will appear.

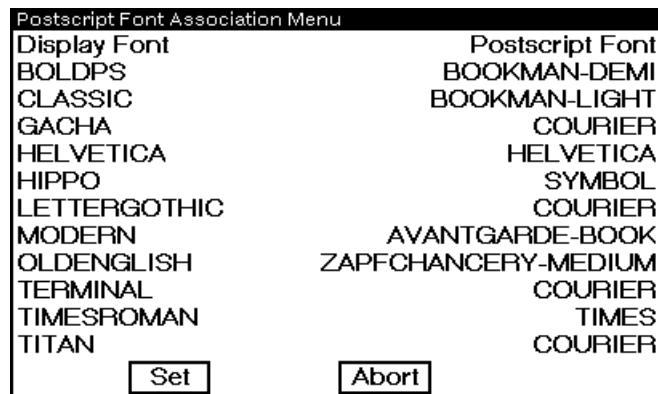


Figure 16-25. **PostScript Font Association** Menu showing the display fonts on the left and the PostScript printer fonts on the right.

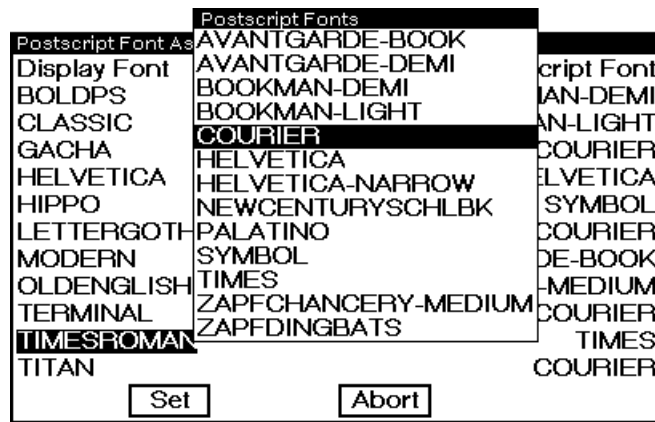


Figure 16-26. **PostScript Font Association** Menu with the **TIMESROMAN** display font selected and about to be set to correspond to the **COURIER** PostScript printer font.

PostScript Short Edge Shift

Is the distance (in points) to shift the image perpendicular to the short edge of the paper. A positive value gives a shift upward in portrait mode, and to the right in landscape mode. A negative value shifts the image down in portrait and to the left in landscaped mode.

The default value is 0.

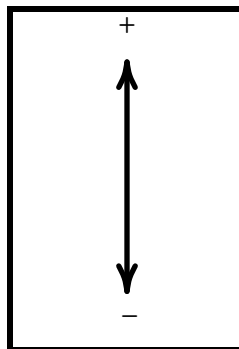


Figure 16-27. Short Edge Shift in Portrait Mode

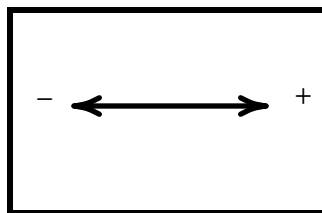


Figure 16-28. Short Edge Shift in Landscape Mode

PostScript Short Edge Points

Indicates the printable region of the page, in points, along the short edge of the paper. It should be adjusted to allow for any short edge shifts of the image.

The default value is 576 points or 8 inches.

PostScript Long Edge Shift

Is the corresponding variable for shifts perpendicular to the long edge of the paper. A positive value here gives a shift to the right in portrait mode and downward in landscape mode.

The default value is 0.

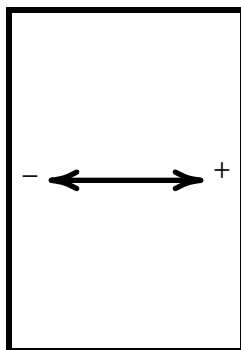


Figure 16-29. Long Edge Shift in Portrait Mode

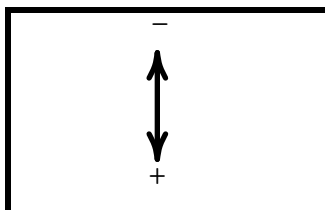


Figure 16-30. Long Edge Shift in Landscape Mode

PostScript Long Edge Points

Indicates the printable region of the page, in points, along the long edge of the paper. It should be adjusted to allow for any long edge shifts of the image.

The default value is 786.24 or 10.92 inches.

Note: The AST TurboLaser PS has an imageable area on the page which is a different size than that of the Apple LaserWriter. The values of **PostScript Short Edge Points** and **PostScript Long Edge Points** for the AST are 575.76 and 767.76, respectively.

Landscape Mode

Indicates whether the default orientation of output files is landscape.

The default value is No.

Landscape Text Mode

Indicates if the printing of text files should force the orientation of output files to landscape.

The default value is No.

Bitmap Scale

Specifies an independent scale factor for display of bitmap images, window hardcopies, for example. Values less than 1 reduce the image size, i.e., a value of 0.5 gives a half size bitmap image. The position of the scaled bitmap still has the same lower-left corner, i.e., the scaled bitmap is not centered in the region of the full size bitmap image.

The default value is 1.

Note: Setting bitmap scale to 0.96, instead of 1, will give cleaner bitmap images on a 300 dpi printer. This corrects for the 72 ppi imagestream vs. the 75 dpi printer, using 4x4 device dots per bitmap pixel. Also, values of 0.24, 0.48 and 0.72, instead of 0.25, 0.5 and 0.75, will also give cleaner images for reduced size output. In general, use integer multiples of 0.24 for a 300 dpi printer.

Texture Scale

Specifies an independent scale for the display of bit map textures. The value represents the number of printer bits per screen pixel. The default value is 4, which represents each pixel of the texture as a 4x4 block, so that textures are approximately the same resolution as on the screen for 300 dpi output devices, such as the Apple Laserwriter.

The PostScript package extends the allowed representations of a texture, beyond 16-bit integers and 16x16 bit maps, to any square bit map. (If the bit map is not square, its longer edge is truncated from the top or right to make it square.) Use this feature with caution, as large bit map textures, or sizes other than multiples of 16 bits square, require large amounts of storage in the PostScript interpreter, and can cause errors when printing.

Image Size Factor

Specifies an independent factor to change the overall size of the printed image. This resizing affects the entire printed output. Values greater than 1 enlarge the printed image, and values less than 1 reduce it. The **Bitmap Scale** parameter does not consider the **Image Size Factor** when determining the scale factor for a bit map.

[This page intentionally left blank]

17. ERROR RECOVERY AND KNOWN PROBLEMS

NoteCards on the Sun Workstation has an error handling system which includes:

- The break window error system
- A diagnostic program, URAID, which handles emulator errors

Occasionally, you may encounter SunOS error messages. Refer to your Sun documentation set for recovery procedures when these errors occur.

Known problems with the NoteCards system are discussed at the end of this chapter.

System Status, Aborting Operations, and Spawning a New Mouse

NoteCards provides a number of keyboard combinations for enquiring about system status and aborting operations.

System Status

Typing CONTROL-T prints status information on the type-in process. The type-in process is the process which has the flashing caret cursor. This is of minimal usefulness as the type-in process is usually just sitting waiting for you to type characters in, and the messages are aimed a people extending the NoteCards system. However, if you need to know whether the system is still active, type CONTROL-T to force the system to give you the status on one process.

Aborting Operations

Pressing the STOP key will frequently abort the process you want it to. However, as the system runs multiple processes, the process which receives the stop command may not be the one you intended to receive it.

To see a list of all the currently running processes, in menu format (as shown in Figure 17-1), type CONTROL-G.

```
Interrupt which process?  
[Spawn Mouse]  
OLDMOUSE *run  
TEdit *tty  
MOUSE  
TEdit#3  
DIGI-CLOCK  
DIRECTORY CONNECTOR  
TEDIT-KILLER#2  
ERIS#LEAF  
\10MBWATCHER#2  
\NSGATELISTENER  
\PUPGATELISTENER  
\TIMER.PROCESS  
BACKGROUND
```

Figure 17-1. **Interrupt Which Process?** Menu Showing All Processes Currently Running

Selecting one of these processes will cause a break to occur in that process. You can then uparrow, ^, out of the break to abort the process. If you decide you have broken the wrong process, you can type **ok** to have the process resume where it left off.

```
INTERRUPT
Interrupt below \TEDIT,HARDCOPY,DISPLAYLINE,
OLDMOUSE/118(debug)ok
```

Figure 17-2. Break caused by CONTROL-G (OLDMOUSE process Resumed with **ok** Command)

Because there can be so many processes, TEdit processes in particular (there is one TEdit process for every TEdit window including the expanded-menu command windows), it is important to know which process you want to break to abort an operation. Generally, if you start an operation with the mouse, it either runs under the MOUSE or OLDMOUSE process, which is deleted the moment the operation finishes. If you see an OLDMOUSE process, this is usually the process you want to break.

There are exceptions to the OLDMOUSE rule. Performing operations from the FileBrowser causes a process with an "FB-" prefix to be spawned. All operations on the TEdit expanded menus are run in the main TEdit process associated with that edit window. If you have one TEdit window open on your screen and one TEdit expanded menu open, you will have two TEdit processes running: TEdit, which is the process associated with the main TEdit window, and TEdit#2, which is the process associated with the expanded menu.

Killing the wrong process can cause you to lose information. USE WITH CAUTION!

Spawning a New Mouse

If by some chance you do manage to kill your mouse process off, it is easy to start a new one. Type CONTROL-G and choose the **[Spawn Mouse]** option on the "Interrupt which process?" Menu. **[Spawn Mouse]** is always the very top option. If you create several new mouse processes, the system removes all but one mouse process.

Break Windows

Occasionally, when performing certain operations like accessing a file which has been moved to another directory, deleted, or renamed, or trying to create a font which does not exist, NoteCards enters a break. When this happens, NoteCards provides you with information about what caused the break, and asks you what to do (Figure 17-3).

```
INTERLISP-ERROR
In ERROR:
FONT NOT FOUND (coerced to)
(HELVETICA 33 (BOLD REGULAR REGULAR
) 0 DISPLAY)
103: ↑
```

Figure 17-3. Break Window Caused by a Font-Not-Found Error

In cases like this, you should first determine if the cause of the break is a simple error. In Figure 17-3, for instance, there is no Helvetica 33 bold font. There is, however, a Helvetica 32. The appropriate action in this case would be to up arrow out of the break window by typing a ^, and reexecuting the operation with the correct font.

Another possibility is that the font directories may have been moved and the system may be looking in the wrong place for the fonts. Here again, you would up arrow out of the break window by typing a ^, edit the initialization file (see Appendix E, Initialization Files) to insert the correct values for DISPLAYFONTDIRECTORIES, load the initialization file, and then retry the operation.

If the break is caused by a missing file, you should up arrow out of the break window by typing a ^, find the current location of the file, and retry the operation with the new path and file name.

If you are unable to uparrow out of a break, or the system is behaving strangely, you should save all your work by closing your notefiles, and saving your TEdit and Sketch files, perform a **Flush image & Exit**, and reload your system. **Flush image & Exit** is a submenu off **Logout** on the background menu.

Lisp Errors

Errors While Running NoteCards

The following errors may occur running NoteCards on the Sun Workstation.

ERROR MESSAGE	CAUSE
File access timed out	Occurs when you try to access a file on a remotely mounted file system or NFS service that is down.
File too large	Self-explanatory.
Too-Many-Files-Open	Occurs when you exceed one of the following: <ul style="list-style-type: none"> • SunOS open file limit (see Chapter 4, System Use Issues) • System file resources while writing a sysout (using the Logout command from the background menu)
Nonexistent directory	Occurs when you try to connect to a nonexistent directory.
No-Such-Directory	Occurs when you try to connect to a nonexistent directory.
Connection timed out	Self-explanatory.
Bad Host Name	Self-explanatory.

Xerox Workstation-Specific Errors

These Xerox workstation-specific errors may occur if certain functions are inadvertently used on the Sun Workstation.

ERROR MESSAGE	CAUSE
Floppy: No floppy drive on this machine.	Self-explanatory.

Device error: {FLOPPY}	Occurs when you try to access a floppy device while running on the Sun Workstation.
Wrong machinetype	Occurs when functions controlling Xerox disk drive device-specific behavior are entered when running on a Sun.

Virtual Memory Errors

ERROR MESSAGE	FUNCTION RESPONSIBLE
File-System-Resources-Exceeded	Logout, Save image & Exit, Save VM
Protection-Violation	Logout, Save image & Exit, Save VM
File-Wont-Open	Logout, Save image & Exit, Save VM

URAIID ---

The NoteCards system normally operates as a self-contained environment. In some unusual circumstances NoteCards may encounter a situation from which it cannot recover. In this case, when an unrecoverable emulator error is encountered, the emulator halts and enters into a small debugger called URAID. URAID allows you to inspect memory, or to look inside the sysout file, and attempt to recover from the error.

If you produce the same type of error condition in NoteCards on a Xerox workstation as you did on a Sun Workstation, you get an MP error instead of a URAID error.

Entering URAID

Normally, the emulator automatically enters URAID when an unrecoverable emulator error occurs. If the system freezes and will not let you regain control, you can throw the system into URAID by simultaneously holding down the SHIFT, CONTROL and DELETE (L10) keys or the SHIFT, CONTROL, and NEXT (ALTERNATE) keys.

URAIID Commands

URAIID has a few simple commands which you can use to attempt error recovery. All URAID commands are case sensitive.

- h** Hard Reset. Attempts to recover by resetting the Lisp stack. Quits URAID and causes NoteCards to resume execution. This command should not be used unless you are sure that execution can be resumed.
- e** Exit to SunOS. NoteCards will end. If you are going to call Customer Support, call *before* you give this command. After giving this command, nothing is recoverable.
- q** Quits URAID and returns to NoteCards.

Note: An error may occur while the NoteCards system is running uninterruptably. The following message signals this error:

```
Error in uninterruptable system code -- ^N to continue into
error handler
```

Disregard the ^N command; it is not supported by URAID. Use the **q** command to continue.

Other Fatal Error Conditions

Occasionally, other emulator, operating system, or system administration errors may occur from which the URAID program cannot recover. Such error conditions include a process dying, the emulator going into an infinite loop, the keyboard being lost, or the system freezing up.

If any of these emulator errors occur, use the UNIX kill command to kill the `lde` process.

System Error Conditions

The following are error messages generated by SunOS. For complete information on these error messages, see the *SunOS Reference Manual*, Intro(2).

ERROR MESSAGE	DESCRIPTION
:0 Unused	
:1 EPERM	Not owner
:2 ENOENT	No such file or directory
:3 ESRCH	No such process
:4 EINTR	Interrupted system call
:5 EIO	I/O error
:6 ENXIO	No such device or address
:7 E2BIG	Arg list too long
:8 ENOEXEC	Exec format error
:9 EBADF	Bad file number
:10 ECHILD	No children
:11 EAGAIN	No more processes
:12 ENOMEM	Not enough core
:13 EACCES	Permission denied
:14 EFAULT	Bad address
:15 ENOTBLK	Block device required
:16 EBUSY	Mount device busy
:17 EEXIST	File exists
:18 EXDEV	Cross-device link
:19 ENODEV	No such device
:20 ENODIR	Not a directory
:21 EISDIR	Is a directory
:22 EINVAL	Invalid argument
:23 ENFILE	File table overflow
:24 EMFILE	Too many open files
:25 ENOTTY	Not a typewriter
:26 Unused	
:27 EFBIG	File too large
:28 ENOSPC	No space left on device
:29 ESPIPE	Illegal seek
:30 EROFS	Read-only file system
:31 EMLINK	Too many links
:32 EPIPE	Broken pipe
:33 EDOM	Math argument
:34 ERANGE	Result too large
:35 EWOULDBLOCK	Operation would block
:36 EINPROGRESS	Operation now in progress
:37 EALREADY	Operation already in progress
:38 ENOTSOCK	Socket operation on non-socket

:39 EDESTADDRREQ Destination address required

ERROR MESSAGE	DESCRIPTION
:40 EMSGSIZE	Message too long
:41 EPROTOTYPE	Protocol wrong type for socket
:42 ENOPROTOPT	Bad protocol option
:43 EPROTONOSUPPORT	Protocol not supported
:44 ESOCKTNOSUPPORT	Socket not supported
:45 EOPNOTSUPP	Operation not supported on socket
:46 EPFNOSUPPORT	Protocol family not supported
:47 EAFNOSUPPORT	Address family not supported by protocol family
:48 EADDRINUSE	Address already in use
:49 EADDRNOTAVAIL	Can't assign requested address
:50 ENETDOWN	Network is down
:51 ENETUNREACH	Network is unreadable
:52 ENETRESET	Network dropped connection on reset
:53 ECONNABORTED	Software caused connection abort
:54 ECONNRESET	Connection reset by peer
:55 ENOBUFS	No buffer space available
:56 EISCONN	Socket is already connected
:57 ENOTCONN	Socket is not connected
:58 ESHUTDOWN	Can't send after socket shutdown
:59 Unused	
:60 ETIMEDOUT	Connection timed out
:61 ECONNREFUSED	Connection refused
:62 ELOOP	Too many levels of symbolic link
:63 ENAMETOOLONG	File name is too long
:64 EHOSTDOWN	Host is down
:65 EHOSTUNREACH	No route to host
:66 ENOTEMPTY	Directory not empty
:67 Unused	
:68 Unused	
:69 EDQUOT	Disc quota exceeded
:70 ESTALE	Stale NFS file handle
:71 EREMOTE	Too many levels of remote in path
:72 ENOSTR	Not a stream device
:73 ETIME	Timer expired
:74 ENOSR	Out of stream resources
:75 ENOMSG	No message of desired type
:76 EBADMSG	Not a data message
:77 EIDRM	Identifier removed

Known Problems

Printing Browsers

Printing Browsers which contain dashed lines may cause your printer to hang.

Notefile Indicator Window

Occasionally, after you have reshaped a card, a piece of the text in the notefile indicator window will be garbled or broken. To redisplay the text in this window, place the point of the mouse cursor in the notefile indicator window, hold down the right mouse button, and select the **Redisplay** option from the window menu which pops up.

Fonts

The PostScript Zapfchancery font does not work at this point in time.

TEdit

When a TEdit window is positioned such that part of the display region of the window is off the screen, TEdit will display some lines of text in a garbled fashion. Simply move the window so that the entire extent of the window is on the screen or redisplay the window using the **Redisplay** option on the title-bar right-button mouse menu.

[This page intentionally left blank]

APPENDIX A - NOTEFILE CONCEPTS

This document provides a brief explanation of the structure of notefiles. It also describes how checkpointing, aborting, and recovering a notefile after a crash work. Finally, it describes the use of **Inspect & Repair** to repair a notefile with inconsistent links. For additional information see Appendix C, Notefile Inspector.

Background Concepts

Unique Identifiers: UIDs

All objects in NoteCards (i.e., cards, links, notefiles) are assigned a unique identifier called a UID. Each UID is a 112-bit number that is guaranteed to be unique across all time and space. UIDs are used in many places in NoteCards as keys for indexing and retrieving cards, links, and notefiles.

Card Parts

For storage purposes a note card is decomposed into four independent parts: contents, title, property list, and links. Each of these parts is stored separately in the data area of the notefile. This is discussed in the Notefile Structure section below. When a card is saved, only those card parts that have changed are rewritten in the notefile.

The contents of the card are stored on the notefile in a manner appropriate to its type. Thus a Text card's content is a text stream and is written on the notefile exactly the way TEdit writes out text streams (i.e., text followed by "looks" information). In contrast, all titles are stored as strings and all property lists as standard Lisp lists. Storage of Links is described in the last section in this chapter.

Notefile Structure: Index and Data Areas

A notefile consists of three parts, a header, an index, and a data area. The header and the index are fixed in size for each notefile. The data area follows the index area and grows as cards are added to or modified in the notefile.

The notefile header contains the following information about the notefile: its UID, a number identifying its format, the checkpoint pointer, the size of the index, and a pointer to the next available index entry. If the notefile header is destroyed, it cannot be automatically reconstructed. Careful hand manipulation of the notefile by a NoteCards wizard is required to recover a notefile with a bad header.

The structure of the index and data area is shown in Figure A-1 .

The index contains a fixed number of index entries. Each index entry that is in use, contains information for one of the cards in the notefile. Specifically, an index entry contains 5 fields: a status character, the card's UID, and 4 pointers. The status character specifies whether the index entry is free (not in use) or contains information for an active or a deleted card. If the index entry is not free, the UID field contains the UID of the card referred to by this index entry and the four pointer fields contain the location in the data area of the 4 parts of its card: contents, title, links and property list.

The number of index entries is fixed at notefile creation time. The default is 1000 entries. The number of index entries is automatically doubled by the notefile compactor if 75% of the entries are used. The compactor also frees (i.e., makes unused) all of the index entries that refer to deleted cards. In normal operation, NoteCards prints a warning whenever more than 90% of the index entries in a notefile are used. At this point, the notefile should be compacted to increase the index size.

The data area contains the actual information about the card. Whenever you change, say, the title of a card, the new title is written at the end of the data area. The title pointer in the card's index entry is then updated to point to this new location in the data area. Thus, in general, a notefile's data area grows every time any part of any card is changed.

The old information, now somewhere in the middle of the data area, is not removed. However, it is no longer directly accessible because there is no index entry that points to it. Thus, for most purposes this old information can be considered "dead space" in the notefile. The notefile compactor rewrites the notefile, eliminating all such dead space.

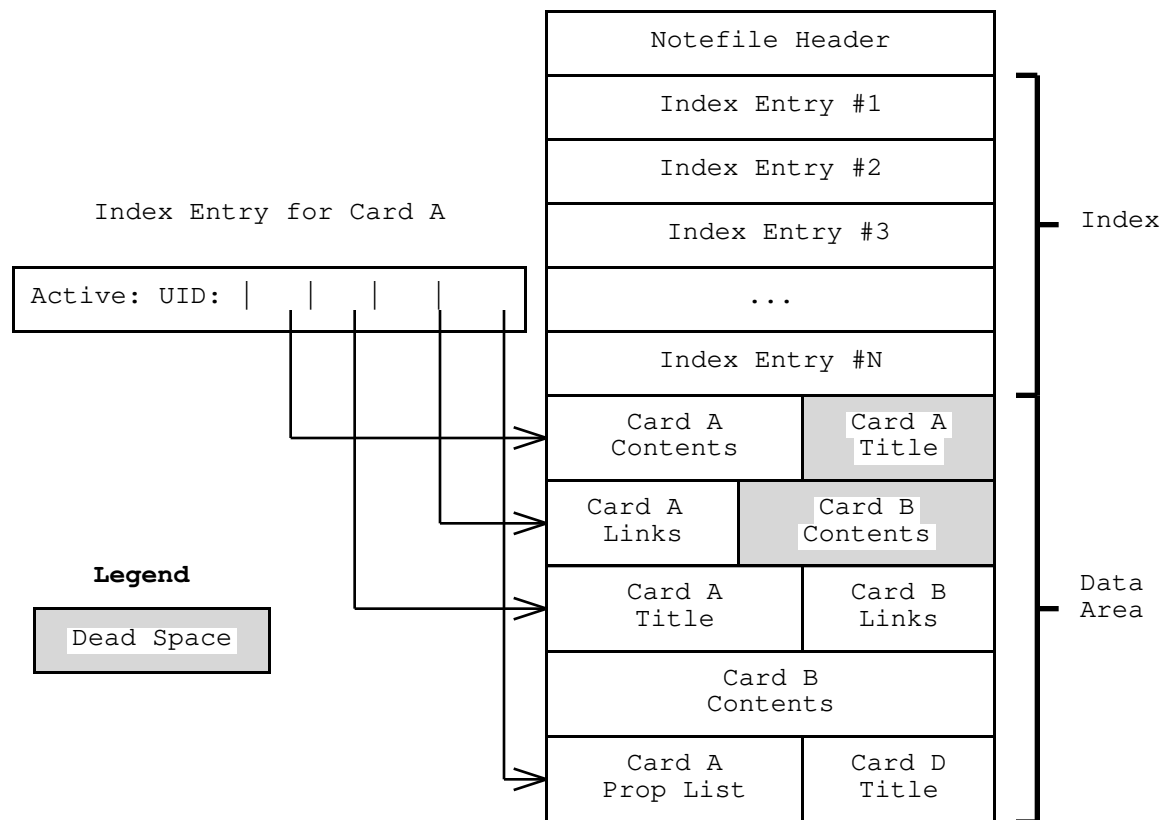


Figure A-1. Structure of a Notefile

As long as the notefile has not been compacted, all the old information can be accessed (and made to be "current") via the notefile Inspect and Repair facility. Inspect and Repair does this by ignoring the index and parsing the entire data area to produce a listing of all the information (both current and old) about a card that is stored on the data area. See the Inspect and Repair manual for more information.

Notefile Checkpointing

As long as a notefile is open, its index area is cached in memory. When a card part is saved, the card part's information is written to the end of the data area but the card's actual index entry on the notefile is not updated with this new location. Instead, the appropriate pointer in the card's in-memory index entry is updated. Thus, the index in the notefile continues to point to the old information in the data area, while the in-memory index points to the new information.

Thus while a notefile is open, its current state is distributed between the actual notefile and information cached in memory. The current index is cached in memory. For cards open on the screen (or cached in memory from the programmer's interface), the "current" card part information is contained in an in-memory cache. For all other cards, the current information is contained in the data area of the notefile pointed to by the in-memory index entry.

If a machine crashes while a notefile is open, the information cached in memory is lost. A crash not only discards changes made to cards on the screen, but it also leaves the information stored on the notefile in an inconsistent state. For example, the index on the notefile may point to old information in the data area. This occurs because the new information (e.g., a new title) is written to the data area but only the in-memory index pointers (which are lost in the crash) have been updated to point to the new information.

Checkpointing forces all of the in-memory information to be written onto the notefile. Specifically, checkpointing causes all open cards to be saved and the in-memory index to be written to the notefile's index area. Thus, immediately after checkpointing, the notefile itself contains its current (and consistent) state. If the machine were to crash at this point, no information would be lost and the notefile would be consistent.

Checkpointing also writes a **checkpoint pointer** onto the notefile header. The checkpoint pointer contains the location of the end of the data area (i.e., the end of the notefile) at the time the checkpoint is done.

As the notefile is used after the checkpoint, information is written in the data area past the checkpoint pointer but only the in-memory index entries are updated to point to this information. The on-file index entries still point to the information in the data area *before* the location referenced by the checkpoint pointer. Thus, a consistent notefile can be constructed from the index area and all of the information in the data area located before the checkpoint location. This is essentially the notefile as it was at the time of the last checkpoint. (Note: one small exception is that changes to a card's size on the screen are actually written in the middle of the data area rather than at the end. Thus, truncating a notefile to its checkpoint location cannot "undo" the reshaping of a card.)

When opening a notefile after a crash, the system will insure that the notefile is in a consistent state. It does so by truncating the data area to the last checkpoint location, saving the truncated information if requested by the user. This leaves the notefile in the state it was during the last checkpoint before the crash.

Aborting a notefile does the same thing. It truncates the data area to the last checkpoint location, thereby eliminating all changes made to the notefile since the last checkpoint. It also discards the in-memory index. Thus, the notefile is left in the exact state it was after the last checkpoint.

Finally, note that notefile Close forces a checkpoint. Therefore, aborts and recovery after crashes actually restore the NoteFile to its state as of the last user-initiated checkpoint or close.

Storing Links and Repairing Notefiles with Inconsistent Links

The links card part is divided into three subcomponents: to-links, from-links, and global links. The to-links is a list of all links whose Source is the given card (i.e., that point from the card to some other card). The from-links is a list of links whose Destination is the given card (i.e., that point from some other card to the given card). Finally, the global links is a subset of the to-links that includes only the Global links originating in the given card.

Given this scheme, every link is stored on the notefile in three different places. First, if the link is Local it is stored inside the link icon which in turn is inside the content part of the link's source card. If the link is global, it is stored in the global links subpart of its source card. Second, the link is stored in the to-links list of its source card. Third, the link is stored in the from-links list of its destination card.

These three records of the same link occasionally get out of synch, resulting in an inconsistent notefile. There are a number of symptoms of such inconsistency. For example, the **ShowLinks** display for card may indicate that the card is a destination for a link from some source card X while the ShowLinks display for X does not include a ToLink to that destination card. Occasionally, inconsistent links will also result in link icons that contain the words "DELETED" or "FREE" when displayed on the screen. This usually means that the card at one end of a link was deleted, but somehow the links of the card at the other end were not updated. Such link icons cause NoteCards to break when you try to follow them.

One function of the NoteCards **Inspect & Repair** facility is to resynchronize the three records for all links in the notefile. The inspector's third phase rebuilds the links as follows. First it removes all to- and from-links for every card. Then it reads the contents for each card and recreates to-links and from-links by looking at the links found inside the link icons in the card's content and in its global links list. In addition, the links rebuilder phase of the notefile inspector can rebuild filebox contents from cards pointed to by the filebox, and the set of all link types from the list of all links.

[This page intentionally left blank]

APPENDIX B - NOTECARDS PROGRAMMER'S INTERFACE

This appendix describes a facility whereby users with some programming know-how can obtain a software interface to NoteCards in Interlisp. In this way, they can create and modify Notefiles, cards and links under program control.

The functions described below are divided into eight groups:

1. NoteFile Creation and Access
2. Creating and Accessing NoteCard Types
3. Creating NoteCards and FileBoxes
4. Accessing NoteCards and FileBoxes
5. Creating and Accessing Links
6. Creating and Accessing Link Labels
7. Customizing the NoteCards Interface
8. Handy Miscellaneous Functions

NoteFile Creation and Access

Some of the following functions accept filename arguments, some take notefile object arguments and some can take either. In any case, if the function accepts a filename, the `.notefile` suffix will be attached if not already present.

(NCP.CreateNoteFile <Filename><quietFlg>)

If <Filename> is not already a notefile, then creates a notefile with name "<Filename>.NoteFile", and returns this filename which can later be passed to NCP.OpenNoteFile. If <quietFlg> is non-nil, then communicative messages are held to a minimum.

(NCP.OpenNoteFile <NoteFileOrFilename> <don'tCreateFlg> <convertw/oConfirmFlg> <quietFlg> <menuPosition> <readOnlyFlg> <Don'tCreateInterfaceFlg>)

<NoteFileOrFileName> can be either a notefile object or a file name. Opens notefile, returning resultant notefile object if successful, else nil. If <don'tCreateFlg> is non-nil, then a new file is not created if the given one doesn't exist. If <convertw/oConfirmFlg> is non-nil, then if needed, the file is converted to the most recent format without user confirmation. If <quietFlg> is non-nil, then communicative messages are held to a minimum. If <menuPosition> is non-nil, then it should be a position at which to bring up the notefile main menu icon. If <readOnlyFlg> is non-nil, then the notefile is opened for read only. If <Don'tCreateInterfaceFlg> is non-nil, then don't bring up a control panel menu for the notefile.

(NCP.CloseNoteFiles <NoteFilesOrT> <quietFlg>)

If <NoteFilesOrT> is a list (or a single notefile), then close all open notefiles on that list (or the single one). If T, then close all open notefiles. If <quietFlg> is non-nil, then communicative messages are held to a minimum.

(NCP.CheckpointNoteFiles <NoteFilesOrT> <quietFlg>)

If <NoteFilesOrT> is a list (or a single notefile), then checkpoint all open notefiles on that list (or the single one). If T, then checkpoint all open notefiles. In case of a system crash or abort, the notefile can be recovered to the last checkpoint. Note that closing a notefile does a checkpoint. <quietFlg> non-nil will keep messages to a minimum.

(NCP.AbortNoteFiles <NoteFilesOrT> <Don'tConfirmFlg> <quietFlg>)

If <NoteFilesOrT> is a list (or a single notefile), then abort all open notefiles on that list (or the single one). If T, then abort all open notefiles. Aborting a notefile closes it and scraps all work since last checkpoint or successful close. <quietFlg> non-nil will keep messages to a minimum. <Don'tConfirmFlg> non-nil will prevent the asking of user for confirmation before throwing away work since last checkpoint.

(NCP.CompactNoteFile <fromNoteFileOrFilename> <toFilename> <inPlaceFlg>)

<fromNoteFileOrFilename> can be either a notefile or a file name. It is compacted, usually recovering space. If <inPlaceFlg> is non-nil, then <toFilename> is ignored and the compaction is in place. Will close <fromNoteFileOrFilename> if it's currently open.

(NCP.CompactNoteFileInPlace <NoteFileOrFilename>)

Compacts <NoteFile> in place, replacing the old version. Equivalent to (NCP.CompactNoteFile <NoteFile> NIL T).

(NCP.NumCardSlotsRemaining <NoteFile>)

Returns the total number of cards that can be created in <NoteFile> before its index must be expanded. <NoteFile> should be an open notefile.

(NCP.ExpandNoteFileIndex <NoteFile> <numNewCardSlots> <QuietFlg>)

Causes <NoteFile> (which should be a valid notefile) to be checkpointed and then have its index expanded in place to make room for <num NewCardSlots> new cards. If <NoteFile> is currently closed, then it is opened, expanded and then closed immediately.

(NCP.RepairNoteFile <NoteFileOrFilename> <readSubstancesFlg>)

Runs the Inspect&Repair facility on <NoteFileOrFilename>. It must *not* be currently open. If <readSubstancesFlg> is non-nil, then the inspector will check the contents of card substances rather than the simpler check of substance length. This means the first phase of the inspector takes MUCH longer so use with care. See the documentation on the Inspect&Repair facility for more information.

(NCP.DeleteNoteFile <NoteFileOrFilename> <Don'tConfirmFlg> <quietFlg>)

Deletes the <NoteFileOrFilename> notefile. Must not be open. <quietFlg> non-nil will keep messages to a minimum. <Don'tConfirmFlg> non-nil will prevent the asking of user for confirmation before deleting.

(NCP.NoteFileFromFileName <Filename>)

Returns the notefile corresponding to Filename if any, else nil.

(NCP.FileNameFromNoteFile <NoteFile>)

Returns the full name of the given notefile.

(NCP.NoteFileMenu <NoteFile>)

Returns the main menu for <NoteFile>.

(NCP.OpenNoteFileP <NoteFile>)

Returns non-NIL if <NoteFile> is a currently open notefile.

(NCP.ValidNoteFileP <NoteFile>)

Returns non-NIL if the notefile is a valid notefile.

(NCP.NoteFileClosingP <DontCheckForAbortFlg >)

Returns non-NIL if it has been called from under a close notefile operation. If DontCheckForAbortFlg is NIL, aborting the notefile counts as closing it. If DontCheckForAbortFlg is non-NIL, aborting the notefile is considered a different operation, and the function will return NIL.

(NCP.ListOfOpenNoteFiles)

Returns a list of all currently open notefiles.

(NCP.CheckOutNoteFile <fromFilename> <toFilename>)

Copies <fromFilename> to <toFilename> unless <fromFilename> is locked. If successful, creates a lock file in <fromFilename>'s directory. The name of the lock file is formed by concatenating the atom LOCKFILE onto <fromFilename>.

(NCP.CheckInNoteFile <fromFilename> <toFilename>)

Check lock file for <toFilename>. If none, then just copy <fromFilename> to <toFilename>. If there is one and it's owned by us, then do the copy and remove the lock file. If there is a lock file owned by someone else or if date of <toFilename> is more recent than date of lock file, then print a message and do nothing.

Creating and Accessing NoteCard Types

These functions give the user access to the NoteCard user-defined types facility. (In the functions below, card type arguments should be atoms.) For example, one of the simplest card types commonly created is what we call a "form" card. At card create time, it adds certain predefined text to the card's contents. Otherwise, form cards behave just like Text cards. Figure B-1 shows the code needed to implement a sample form card type. For a full explanation of this facility, see the NoteCards Types Mechanism documentation.

FooForm.AddFooFormType

```
(LAMBDA NIL (* rht; "10-Jul-86 11:41")

(* * This creates the FooForm card type.)

(NCP.CreateCardType (QUOTE FooForm)
  (QUOTE Text)
  (QUOTE ((MakeFn FooForm.MakeFn)))
  (QUOTE ((DisplayedInMenuFlg T))))
```

FooForm.MakeFn

```
(LAMBDA (Card Title NoDisplayFlg) (* rht; "10-Jul-86 11:55")

(* * The MakeFn for the FooForm card type.
Note the call to the Text card's makefn.)

(PROG1 (NCP.ApplySuperTypeFn MakeFn Card Title NoDisplayFlg)
  (NCP.CardAddText Card (CONCAT
    "Some initial text for the FooForm card."
    (CHARACTER 13)
    ">>field1<<"
    (CHARACTER 13)
    "etc."))))
```

Figure B-1: Sample card type definition

(NCP.CardTypes)

Returns list of all currently defined NoteCard types.

(NCP.CardTypeP <type>)

Returns non-nil if <type> is an existing NoteCard type.

(NCP.CardTypeFns)

Returns a list of the valid Fns fields for NoteCard types. Currently, these include: MakeFn, EditFn, QuitFn, GetFn, PutFn, CopyFn, MarkDirtyFn, DirtyPFn, CollectLinksFn, DeleteLinksFn, UpdateLinkIconsFn, InsertLinkFn and TranslateWindowPositionFn.

[The following hooks should some day become legitimate Fns or Vars fields for NoteCard types. Currently, they are properties placed on the card type's atom.

WhenSavedFn

If you need to have certain functionality happen just before a card's contents is saved to the notefile, then make a function be the value of the WhenSavedFn prop of the card type atom. It is called whenever a card's contents is saved.

WhenDeletedFn If you need to have certain functionality happen just before a card is deleted, then make a function be the value of the WhenDeletedFn prop of the card type atom. It will be called whenever a card is deleted. If the function returns 'ABORT', then the deletion of the card will be aborted.

LinkIconLeftButtonFn One may now specify an operation other than TraverseLink to be used when left buttoning in link icons. Just put the property "LinkIconLeftButtonFn" with the value of a function on the (destination card's) card type atom. The function will get called with two args, the destination card and the window containing the link icon. (This is similar to the ExtraLinkIconMenuItems card type property consulted by the link icon middle button code.)

AttachedBitMapFn You may now supply a function to calculate the attached bitmap for a card type. This function should be the value of the AttachedBitMapFn prop of the card type atom, and it will be passed Card, ScaledHeightToMatch, and Scale. If it exists, it will be applied first in calculating the bitmap to be displayed. If it returns a bitmap, a list of heights and bitmaps will be computed and placed on the LinkIconAttachedBitMap field of the card type. This hook may also be used to calculate the bitmap to be displayed on the right side of a link icon if it is a cross-file link and its attached bitmap is being displayed. In this case, the function should be the value of the AttachedBitMapFn prop of the card type atom 'CrossFileLink'. (For computing the list of heights and bitmaps in an AttachedBitMapFn, see NCP.MakeTypeIconBitMapSet at the end of this section.)

Don'tForceFilingFlg It is possible to turn off forced filing on either the card type level or the card level. To define all cards of a type as not needing filing, you should put the property "Don'tForceFilingFlg" with the value of T on the card type atom. To define an individual card as not needing filing, use the function (NCP.MarkAsNotNeedingFiling <card>) (see Section 8: Handy Miscellaneous Functions).

NewCardPos If present, the value of this property determines where new cards of the type are brought up on the screen, assuming the RegionOrPosition argument now accepted by all card type makefns (passed to NC.DetermineDisplayRegion) is NIL.]

(NCP.CardTypeVars)

Returns a list of the valid Vars fields for NoteCard types. Currently, these include: SuperType, StubFlg, FullDefinitionFile, LinkDisplayMode, DefaultWidth, DefaultHeight, LinkAnchorModesSupported, DisplayedInMenuFlg, LinkIconAttachedBitMap, LeftButtonMenuItems and MiddleButtonMenuItems.

(NCP.CardTypeFnP <fn>)

(NCP.CardTypeVarP <var>)

Returns non-nil if <fn> (<var>) is a valid function (variable) field for NoteCard types, for example, the litatom MakeFn (DefaultWidth). In other words, <fn> (<var>) can serve as the <fn> (<var>) arg to NCP.CardTypeFn (NCP.CardTypeVar).

(NCP.CardTypeFn <type> <fn>)

Returns the <fn> field for <type>. Note that this may be a value inherited at card type creation from <type>'s super type.

(NCP.CardTypeVar <type> <var>)

Returns the <var> field for <type>. Note that this may be a value inherited at card type creation from <type>'s super type.

(NCP.CardTypeSuper <type>)

Returns the super type of <type>. Equivalent to (NCP.CardTypeVar <type> 'SuperType).

(NCP.CreateCardType <TypeName> <SuperType> <FnsAssocList> <VarsAssocList>)

Makes a new NoteCard type with name <TypeName> and super type <SuperType>. Any functions not appearing in <FnsAssocList> or vars not appearing in <VarsAssocList> will be inherited from <SuperType>.

Note that, for now, specializing the FileBox card type is very dangerous and has unknown repercussions.

(NCP.DeleteCardType <TypeName> <DeleteSubTypesFlg>)

Deletes the NoteCard type with name <TypeName>. If DeleteSubTypesFlg is non-NIL, then it recursively deletes all sub-types of <TypeName>. If DeleteSubTypesFlg is NIL, then attempting to delete a type with sub-types is an error.

(NCP.CreateCardTypeStub <TypeName> <SuperType> <FullDefinitionFileName> <FnsAssocList> <VarsAssocList>)

Makes a stub for a new NoteCard type with name <TypeName> and super type <SuperType>. Some subset of the fns and vars can appear in <FnsAssocList> and <VarsAssocList>, however, the full definition will be loaded from <FullDefinitionFileName> the first time an attempt is made to access an undefined field name.

(NCP.ChangeCardTypeFields <TypeName> <FnsAssocList> <VarsAssocList>)

<TypeName> should be an existing NoteCard type. Some subset of the fns and vars should appear in <FnsAssocList> and/or <VarsAssocList>. These will be changed in the card type and then the inheritance mechanism will propagate these changes to any inheriting card types. For example, to make some existing card type FOO appear in the card type menu, do (NCP.ChangeCardTypeFields 'FOO NIL '(DisplayedInMenuFlg T))).

(NCP.ApplyCardTypeFn <CardTypeFn> <Card> <arg1> ...)

This macro applies the card type fn <CardTypeFn> (unevaluated) of the card type of <Card> to <Card> and the other args.

(NCP.ApplySuperTypeFn <CardTypeFn> <Card> <arg1> ...)

This macro applies the card type fn <CardTypeFn> (unevaluated) of the super type of <Card>'s type to <Card> and the other args.

(NCP.IsSubTypeOfP <type1> <type2>)

Returns non-nil if type1 inherits directly or indirectly from type2, that is, type2 can be found somewhere up the SuperType chain from type1.

(NCP.TextBasedP <cardOrType>)**(NCP.SketchBasedP <cardOrType>)****(NCP.GraphBasedP <cardOrType>)**

If <cardOrType> is a card then we use its type. Returns non-nil if that type inherits directly or indirectly from Text, Sketch or Graph respectively.

(NCP.AutoLoadCardType <TypeName>)

<TypeName> is currently undefined. This asks NoteCards to look around for the file containing it and load it. It searches NOTECARDSDIRECTORIES for a file named NC<TypeName>TYPE.

(NCP.LinkIconAttachedBitMap <TypeName> <Size>)

Returns the link icon attached bitmap of size Size for card type TypeName. Default size is determined by the global variable NCP.DefaultLinkIconAttachedBitMapSize, whose initial value is 17.

(NCP.MakeTypeIconBitMapSet <Bitmap> <Heights>)

Returns a list of heights and bitmaps to be used in determining the appropriately sized bitmap for a given link icon. (Typically used when supplying an AttachedBitmapFn - see above.) <Bitmap> will be scaled to all of the heights in <Heights>, which defaults to NC.DefaultLinkIconAttachedBitMapHeights if NIL.

Creating NoteCards and FileBoxes

The following functions create various sorts of cards and boxes within the currently open notefile.

**(NCP.CreateCard <type> <NoteFile> <title> <nodisplayflg> <props>
<parentfileboxes> <otherargs><InterestedWindow> <RegionOrPosition>)**

Creates and returns a card of the given (possibly user-defined) type, with given title, props, and parents. <otherargs> is a possibly nil list of args that will be passed to the MakeCardFn of <type>. Card is initially displayed or not according to value of <nodisplayflg>. Note that actually a top level copy of <props> is used (i.e. (APPEND <props>)). <InterestedWindow> is a window used to attach a prompt window for messages. <RegionOrPosition> is used to position and/or shape the new card.

(NCP.CreateTextCard <NoteFile> <title> <nodisplayflg> <props> <parentfileboxes> <InterestedWindow> <RegionOrPosition>)

Creates and returns a new notecard having type Text. If <title> is non-nil, it is installed as the Notecard's title, otherwise the title is "Untitled." <props>, if non-nil, should be a prop-list of properties and values to be placed on the user property list of the Notecard. If <parentfileboxes> is non-nil, then it should be a list of FileBoxes in which to initially file this card. <InterestedWindow> is a window used to attach a prompt window for messages. <RegionOrPosition> is used to position and/or shape the new card.

(NCP.CreateFileBox <NoteFile> <title> <nodisplayflg> <props> <childcardsboxes> <parentfileboxes> <InterestedWindow> <RegionOrPosition>)

Creates and returns a new Filebox with title <title> (or a gensym'ed name if <title> is nil). It will initially contain child cards and boxes from the list <childcardsboxes> (if that arg is non-nil). If <parentfileboxes> is nil, then the new filebox will be filed in the value of (NCP.GetToBeFiledFileBox). The <props> arg is handled as it was for NCP.CreateNoteCard. <InterestedWindow> is a window used to attach a prompt window for messages. <RegionOrPosition> is used to position and/or shape the new card.

(NCP.CreateBrowserCard <NoteFile> <title> <paramList> <nodisplayflg> <props> <parentfileboxes> <InterestedWindow> <RegionOrPosition>)

Creates and returns a new browser card with given title, props and parents. <paramList> should be a prop list of browser parameters. The properties currently recognized are:

ROOTCARDS	A list of Notecards to serve as roots of the forest or lattice generated by the browser. If omitted or NIL then user is asked to choose root cards.
LINKTYPES	A list of link types to follow when creating the browser. Any label present in the list having the backarrow prefix ("_") represents that link type but in the reverse direction. This list can also contain the atoms ALL or _ALL (<backarrow>ALL) in which case browsing will be done on all links in either the forward or reverse direction. If both ALL and _ALL (<backarrow>ALL) are specified, then links in both directions will be used (generally making a mess).
DEPTH	The depth at which to cut off the browser. This should be a non-negative integer. If NIL or omitted, then will assume no limit. (Currently integers greater than 9 are assumed equivalent to infinity.)
FORMAT	This should be a list of one, two or three elements. The first should be an atom indicating grapher format. The choices are FAST (layed out as a forest, sacrificing screen space for speed), COMPACT (layed out as a forest, using minimal screen space), LATTICE (layed out as a directed acyclic graph, the default), *GRAPH* (layed out as a graph, i.e. virtual nodes are eliminated). The second element of the FORMAT list, if present, should be either HORIZONTAL (the default) or VERTICAL specifying whether the graph is layed on its side or up and down. The

third element, if present, should be the atom REVERSE. This indicates that horizontal graphs should be layed out from right to left instead of left to right and that vertical graphs should be layed out from bottom to top rather than vice versa.

If all of LINKTYPES, DEPTH, or FORMAT are omitted, the user is asked to choose them from a stylesheet. If one or more is specified, even as being NIL, the user is not prompted for them. <InterestedWindow> is a window used to attach a prompt window for messages. <RegionOrPosition> is used to position and/or shape the new card.

(NCP.CreateSketchCard <NoteFile> <title> <nodisplayflg> <props> <parentfileboxes> <InterestedWindow><RegionOrPosition>)

Creates and returns an initially empty sketch/map card having given title, props, and parents. <InterestedWindow> is a window used to attach a prompt window for messages. <RegionOrPosition> is used to position and/or shape the new card.

(NCP.CreateGraphCard <NoteFile> <title> <nodisplayflg> <props> <parentfileboxes> <InterestedWindow><RegionOrPosition>)

Creates and returns an initially empty graph card having given title, props, and parents. <InterestedWindow> is a window used to attach a prompt window for messages. <RegionOrPosition> is used to position and/or shape the new card.

(NCP.MakeDocument <NoteFile> <rootcard> <parametersProplist> <nodisplayflg> <props> <parentfileboxes><InterestedWindow> <RegionOrPosition>)

Creates and returns a Document card starting from <rootcard>. The user may specify new values for the set of parameters for making a document with <parametersProplist>. For example, a value of '(TitlesFromNoteCards Bold ExpandEmbeddedLinks ALL)' for <parametersProplist> would cause these values for the parameters TitlesFromNoteCards and ExpandEmbeddedLinks to be used, and the current defaults for the other parameters will be used. If no <parametersProplist> is provided, the user will be prompted for the parameters with a stylesheet. As usual, the resulting card will have the given props and parents. <InterestedWindow> is a window used to attach a prompt window for messages. <RegionOrPosition> is used to position and/or shape the new card.

(NCP.MakeLinkIndex <NoteFile> <linktypes> <backpointersP> <nodisplayflg> <props> <parentfileboxes> <InterestedWindow> <RegionOrPosition>)

Creates and returns a LinkIndex text card consisting of a sorted record of all instances of links in the current notefile having one of the given link types. <linktypes> can contain the litatoms ALL and/or _ALL (<backarrow>ALL) as well as any particular backwards links. (See the above description of NCP.MakeDocument.) Backpointer links are inserted in the text if <backpointersP> is non-nil. Resulting card will have given props and parents. <InterestedWindow> is a window used to attach a prompt window for messages. <RegionOrPosition> is used to position and/or shape the new card.

Accessing NoteCards and FileBoxes

The following functions provide access to the cards and boxes present in the current notefile. Note that whether a card's window has been brought up on the screen has little or no impact on most of the following functions. If the user changes some field of a card while that card is visible on the screen, then the field will update itself automatically. Thus, users can switch between program-driven and screen-interface-driven modes at will.

Most of the following functions take as first argument a card or filebox. If this does not in fact correspond to an existing card or box, then an error message is printed and nil is returned.

Cards can be displayed, cached or closed. A cached card has its information cached in memory thus saving time (to access) at the expense of space. All cards displayed on the screen are also cached. In almost all cases, users will need only use NCP.OpenCard and NCP.CloseCards. NCP.OpenCard does both caching and displaying while NCP.CloseCards does both undisplaying and uncaching, if necessary.

(NCP.OpenCard <card> <region/position> <typeSpecificArgs>)

Brings up on the screen the given card in the given region or at the given position. If <region/position> is nil, then user is asked to specify position with mouse. <typeSpecificArgs>, if any, will be passed to the card type's EditFn.

(NCP.CloseCards <cardOrListOfCards> <quietFlg>)

Undisplay and uncache all the cards in <cardOrListOfCards>. <quietFlg> non-nil cuts down on messages.

(NCP.DisplayCard <card> <region/position> <typeSpecificArgs>)

If <card> was cached but not displayed, then bring it up on the screen. The <region/position> argument is as in NCP.OpenCard. <typeSpecificArgs>, if any, will be passed to the card type's EditFn.

(NCP.UndisplayCards <cardOrListOfCards> <quietFlg> <writeChangesFlg>)

If any of <cardOrListOfCards> were displayed, then undisplay them. <quietFlg> non-nil cuts down on messages. Normally, changes are not written to the notefile when a card is undisplayed, but only when it is uncached. <writeChangesFlg> non-nil makes changes be written through to the notefile.

(NCP.CacheCards <cardOrListOfCards>)

If any of <cardOrListOfCards> are not currently cached, then cache them.

(NCP.UncacheCards <cardOrListOfCards> <quietFlg>)

If any of <cardOrListOfCards> are cached but not displayed, then uncache them. <quietFlg> non-nil cuts down on messages.

(NCP.CardDisplayedP <card>)

Returns non-nil if given card or box is currently displayed in a window.

(NCP.CardCachedP <card>)

Returns non-nil if given card's information is currently cached.

Most of the following functions leave the card in the same state as it was when they started (except NCP.BringUpCard, which makes it active). Thus, users needing to do several consecutive operations to the same card should consider temporarily caching the card's information via NCP.CacheCards (and then uncaching with NCP.CloseCards).

(NCP.CardType <card>)

Returns the type of <card> or NIL if the card does not exist.

(NCP.ValidCardP <card>)

Returns <card> if <card> exists (hasn't been deleted), otherwise returns NIL. (This is currently a synonym for NCP.CardType.)

(NCP.SameCardP <card1> <card2>)

Returns non-nil if <card1> is the same card as <card2>. Error if either arg is not a valid card.

(NCP.NewCardP <card>)

Returns non-nil if <card> is new; ie. not yet saved in the notefile. Error if the arg is not a valid card.

(NCP.CardBeingDeletedP <card>)

Returns non-nil if <card> is in the process of being deleted; for use with user-defined card types. Error if the arg is not a valid card.

(NCP.CardTitle <card> [<newtitle>])

Returns old title of <card>. If <newtitle> is present, then set <card>'s title to <newtitle>. <newtitle> can be an atom or string. Note, however, that all titles are converted internally to strings by NoteCards.

(NCP.FileCards <cards> <fileboxes>)

Every card or box in <cards> is filed in every box in <fileboxes>. Either arg may be a card object or a list.

(NCP.UnfileCards <cards> <fileboxes>)

Every card or box in <cards> is unfiled from every box in <fileboxes>. Furthermore if <cards> is the litatom ALL, then the boxes in <fileboxes> will be cleared of all children. Similarly, if <fileboxes> is the litatom ALL, then the cards and boxes in <cards> will be unfiled from all their parent boxes. Either arg may be a card object or a list.

(NCP.CardParents <card> <FollowCrossFileLinksFlg>)

Returns list of fileboxes in which <card> is filed. <FollowCrossFileLinksFlg>, if non-nil, causes cross-file links to be followed to recover fileboxes in remote notefiles in which <card> is filed.

(NCP.FileBoxChildren <filebox> <FollowCrossFileLinksFlg>)

Returns list of children of <filebox> in the order in which they appear in the box's textstream. <FollowCrossFileLinksFlg> is as in NCP.CardParents.

(NCP.GetLinks <cards> <destinationCards> <labels> <NoteFile>)

Returns list of all links from any of <cards> to any of <destinationCards> having any label in <labels>. Any of these arguments can be nil. For example, if <destinationCards> is nil, then all links pointing from <cards> to anywhere with a label in <labels> are returned. If both <cards> and <destinationCards> are nil, then <NoteFile> should not be nil and this returns all links in <NoteFile> having a label in <labels>. If all three args are nil, then this is a slow synonym for (NCP.AllLinks <NoteFile>).

(NCP.GetCrossFileLinkDestCard <CrossFileLinkCard> <InterestedWindow> <Don'tOpenDestNoteFileFlg>)

For a given <CrossFileLinkCard>, tries to follow the link to a remote card in the destination notefile and returns that card if found. If <Don'tOpenDestNoteFileFlg> is non-nil, then destination notefile must be open. <InterestedWindow> is an optional window argument whose prompt window is used for messages.

Cross-file link cards are "hidden" cards that serve as placeholders for true cross-file links. That is, a (two-way) cross-file link from card A to card B (in different notefiles) actually consists of 2 links and 2 cross-file link cards. In the source notefile, card A is linked to a cross-file link card AA. AA "knows" about the notefile that contains B and B's UID. Similarly, the destination notefile contains a cross-file link card BB which is linked to card B. Again, the substance of BB contains a filename "hint" for the source notefile as well as the UID for A. One-way cross-file links only contain cross-file link cards in the source notefile. The destination card contains no record that it's been linked to.

(NCP.CardNeighbors <cards> <linkTypes> <FollowCrossFileLinksFlg>)

Return a list of cards each of which is one link away from some card in <cards>. Only links having types in <linkTypes> are considered. Both <cards> or <linkTypes> can be either nil, card objects (or link type atoms) or a list. Null <cards> means that all cards linked to by anyone are returned. <linkTypes> can include link types prefixed with the character '_' (<backarrow>). This means to follow links of that type in the reverse direction. <linkTypes> can also include either or both of the atoms ANY and _ANY (<backarrow>ANY). The former means to include any cards one link away in the forward direction, while the latter causes inclusion of any cards one link away in the backwards direction. <linkTypes>=NIL is equivalent to <linkTypes>='ANY. If <FollowCrossFileLinksFlg> is non-nil, then any links that cross notefile boundaries are followed to their remote destination cards. Otherwise, cross-file links are ignored.

(NCP.CardPropList <card>)

Returns the prop list of the given card.

(NCP.CardProp <card> <propname> [<newvalue>])

Returns old value of property <propname> on <card>'s prop list. If <newvalue> is present, then set <card>'s <propname> property to <newvalue>. (Semantics are analogous to the Interlisp function WINDOWPROP.)

(NCP.CardAddProp <card> <propname> <newitem>)

Adds <newitem> to the list present on the <propname> property of <card>. Returns old value of property. (Semantics are analogous to WINDOWADDPROP.)

(NCP.CardDelProp <card> <propname> <itemToDelete>)

Deletes <itemToDelete> from the <propname> property of <card> if it is there, returning the previous value of that property. If not there, return nil. (Semantics are analogous to WINDOWDELPROP.)

(NCP.CardRegion <card> [<newRegion>])

Returns the region of <card>. This works even if <card> is not currently up on the screen, since the region information is stored on the notefile. If <newRegion> is provided, then the saved region of the card is changed. If the card is currently displayed, then it is reshaped to the new region.

(NCP.CardSubstance <card> [<newSubstance>])

Returns the substance of <card>. For example, returns a textstream in the case that the type of <card> is built on the TEdit text editor. In general, this is what the PutFn of a card type writes down to the notefile and what the GetFn reads in. If a <newSubstance> argument is present, then the substance of the card is replaced and NCP.MarkCardDirty is called.

(NCP.CardAddText <card> <textstr> <loc>)

Adds the text within the string <textstr> to the text card <card>. If <loc> is the litatom START or END, then the text will be placed at the start or end of the card respectively. If <loc> is a number, then it is assumed to be a character count within the card at which to place the new text. If <loc> is NIL, then the text is placed at the current cursor location.

(NCP.ChangeLoc <card> <loc>)

Changes the cursor's location in <card>'s textstream to <loc>. Possible values for <loc> are as described for NCP.CardAddText.

(NCP.DeleteCards <cardOrListOfCards>)

Deletes the given cards and fileboxes from their notefile, or deletes just the one if <cards> is a single card object.

(NCP.CardNoteFile <card>)

Returns <card>'s NoteFile.

(NCP.CardWindow <card>)

Returns <card>'s window if <card> is currently displayed somewhere on the screen.
(The function NCP.WindowFromCard is an alias for NCP.CardWindow.)

(NCP.CardFromWindow <window>)

Returns the card associated with <window>, or NIL if not a notecards window.

(NCP.CardFromTextStream <TextStream>)

Returns the card associated with <TextStream>, or NIL if <TextStream> doesn't belong to some text card.

(NCP.FileBoxP <card>)

Returns non-nil if <card> is a filebox.

(NCP.AllCards <NoteFile>)

Returns a list of all extant cards for the given notefile.

(NCP.CardsOfTypes <CardsOrNoteFile> <Types>)

Returns a list of all cards of the given type (or types) for the given notefile (or list of cards).

(NCP.AllBoxes <NoteFile>)

Returns a list of all fileboxes in the given notefile.

(NCP.MapCards <NoteFile><fn> <collectResultsPredicate>)

Maps down the set of all cards in the current notefile, applying <fn> to each. If <collectResultsPredicate> is non-nil, then for those cards satisfying the predicate, the values of <fn> applied to them are collected.

(NCP.MapCardsOfType <types> <NoteFile> <fn> <collectResultsPredicate>)

This is similar to NCP.MapCards, but only looks at cards whose type appears on <types>. <types> can be a single type or a list of types.

(NCP.ContentsFileBox <NoteFile>)**(NCP.OrphansFileBox <NoteFile>)****(NCP.ToBeFiledFileBox <NoteFile>)**

These functions retrieve the three predefined FileBoxes for the currently open NoteFile. These boxes can be modified (but not deleted) by the user in the same way as any other filebox.

Creating and Accessing Links

Links consist of source card, destination card, link type, display mode and anchoring mode. The new function `NCP.CreateLink` can be used to create any sort of link. We still provide the four functions `NCP.GlobalGlobalLink`, `NCP.LocalGlobalLink`, etc. for those who have grown used to that style.

(NCP.GetLinks <cards> <destinationCards> <labels> <NoteFile>)

See documentation in the previous section.

(NCP.CreateLink <source> <destination> <linkType> <displayMode>)

<source> can be either a card or a list of two elements (<sourceCard> <sourceLoc>). <sourceCard> should be a card to use as the source of the link while <sourceLoc> should be either the atom GLOBAL (in which case a global-to-global link is created) or a Loc directive as described in `NCP.CardAddText` above, that is, an integer or one of the atoms START, END or NIL. This creates and returns a new link with type <linkType>, connecting <sourceCard> to <destinationCard>. For text cards, Loc, if present, designates where to insert the link. If the link is local-to-global, then <displayMode> should be a valid displaymode or NIL. (See description of `NCP.LinkDisplayMode` for the valid values for <displayMode>.) (In the future, Locs for non-text cards will be specifiable. In the far future, we hope to allow local anchoring at the destination end of the link as well as the source.)

(NCP.GlobalGlobalLink <label> <sourceCard> <destinationCard>)

Creates and returns a new link with label <label>, connecting <sourceCard> to <destinationCard>.

(NCP.LocalGlobalLink <label> <sourceCard> <destinationCard> <fromloc> <displaymode>)

Creates and returns a new link with label <label>, connecting from <fromloc> of <sourceCard> card to <destinationCard>. If <displaymode> is non-nil, then the new link is displayed in the given mode. Otherwise the default displaymode for the source card's type is used. See the description of `NCP.LinkDisplayMode` for the valid entries for the <displaymode> arg.

(NCP.GlobalLocalLink <label> <sourceCard> <destinationCard> <toloc>)

Not implemented at this time.

(NCP.LocalLocalLink <label> <sourceCard> <destinationCard> <fromloc> <toloc>)

Not implemented at this time.

(NCP.LinkDesc <link> <followCrossFileLinkFlg>)

Returns list of three items (<label> <sourceDesc> <destinationDesc>) where <label> is the link type and <sourceDesc> and <destinationDesc> have the form (<anchor mode> <card> <loc>). <anchor mode> is either LOCAL or GLOBAL, <card> is the card at this end of the link, and <loc> gives a position in the text of <card> if <anchor type> is LOCAL and <card>'s substance's type is TEXT. If the link is a cross-file link and if <followCrossFileLinkFlg> is non-nil, then the link will be traversed, opening the remote

notefile if necessary to determine information about the source or destination card of the link.

(NCP.LinkDisplayMode <link> [<newdisplaymode>])

Returns old display mode of <link>. If <newdisplaymode> is present, then set <link>'s displaymode accordingly. If non-nil, it can be an instance of the LINKDISPLAYMODE record. Or it can be one of the litatoms Icon, Title, Label, or Both. Finally, it can be a list of three elements (<ShowTitleFlg> <ShowLinkTypeFlg> <AttachBitmapFlg>). Each element can have one of the three values T, NIL, or FLOAT. If a field, say <ShowTitleFlg>, has value FLOAT then the corresponding global parameter (DefaultLinkIconShowTitle, in this case) will be consulted to decide whether or not to display the destination card's title in this icon. (See Section 7 for a description of the global parameters.)

(NCP.CoerceToLinkDisplayMode <thing>)

Returns a LINKDISPLAYMODE record. Thing can be a LINKDISPLAYMODE record, cardtype, card, link, atom, or list. If thing is a LINKDISPLAYMODE record, that record is returned. If thing is a cardtype, the default LinkDisplayMode for that cardtype is returned. If thing is a card, the LinkDisplayMode of that card is returned. If thing is a link, the LinkDisplayMode of that link is returned. If thing is an atom or a list, then the corresponding LinkDisplayMode, as specified under **NCP.LinkDisplayMode** (see above), is returned.

(NCP.LinkType<link> [<newLinkType>])

Returns old linktype of <link>. If <newLinkType> is present, set <link>'s type to <newLinkType>.

(NCP.LinkSource <link>)

Returns the card at the source end of <link>.

(NCP.LinkDestination <link>)

Returns the card at the destination end of <link>.

(NCP.DeleteLinks <links>)

Removes all links in <links> (or just one if <links> is a single link object).

(NCP.ValidLinkP <link>)

Returns non-nil if <link> is a link in the current notefile.

(NCP.SameLinkP <link1> <link2>)

Returns non-nil if <link1> is the same link as <link2>. Error if either arg is not a valid link.

(NCP.AllLinks <NoteFile>)

Returns a list of all existing links in <NoteFile>. (This is equivalent to but faster than (NCP.GetLinks NIL NIL NIL <NoteFile>).)

(NCP.MapLinks <NoteFile> <fn> <collectResultsPredicate>)

Maps down the set of all links in the given notefile, applying <fn> to each. If <collectResultsPredicate> is non-nil, then for those links satisfying the predicate, the values of <fn> applied to them are collected.

(NCP.MapLinksOfType <types> <NoteFile> <fn> <collectResultsPredicate>)

This is similar to NCP.MapLinks, but only looks at links whose type appears on <types>. <types> can be a single type or a list of types.

Creating and Accessing Link Labels

The following functions allow the user to manipulate link labels.

(NCP.CreateLinkType <linkType> <NoteFile> <QuietFlg>)

Creates a new link type with name <LinkType> in <NoteFile> unless there is already one defined by that name. A non-NIL <QuietFlg> suppresses the error message that is normally printed if the link type has already been defined for this notefile.

(NCP.DeleteLinkType <linkType> <NoteFile>)

Deletes the link type <linkType> from <NoteFile>. The link type must exist and must not be the type of any existing link, and it must not be a system-defined link type (e.g. SubBox or BrowserContents).

(NCP.RenameLinkType <linkType> <newLinkType> <NoteFile>)

Changes any links in <NoteFile> having link type <linkType> to have type <newLinkType>. <linkType> must exist and neither <linkType> nor <newLinkType> should be a system-defined type.

(NCP.LinkTypes <NoteFile>)

Returns a list of all existing link types in <NoteFile> including system-defined ones.

(NCP.ReverseLinkTypes <NoteFile>)

Returns a list of the reverse link types for every link type in <NoteFile>. Thus, whereas SubBox would appear in the list returned by NCP.LinkTypes, _SubBox (<backarrow>SubBox) would appear in the list returned by NCP.ReverseLinkTypes.

(NCP.UserLinkTypes <NoteFile>)

Returns a list of all existing user-defined link labels in <NoteFile>.

(NCP.SystemLinkTypeP <LinkType>)

Returns non-nil if <LinkType> is a system link type..

(NCP.ValidLinkTypeP <LinkType> <NoteFile>)

Returns non-nil if <LinkType> is a defined link type for <NoteFile>.

Customizing the NoteCards Interface

There are currently several areas where the user may tailor the NoteCards user interface: the NoteCards Session Icon menus, the left button menu on the title bar of a notefile's menu icon, the middle button menu of a notefile's menu icon, the ShowCards middle button menu on a notefile's menu icon, and the title bar menus of displayed cards and card types.

The Session Icon: The menus associated with this icon may be modified using the following functions:

(NCP.AddSessionIconMenuItem <MenuName> <Item>)

Adds the menu item <Item> to the session icon menu specified by <MenuName> (one of 'Card, 'NoteFile, or 'Other). <Item> should be a complete menu item and may contain subitems in the standard format. For the Card and Other menus, the item can be a normal menu item. For the NoteFile menu, it must be slightly different. Instead of providing an expression to be evaluated as the second part of the menu item (and of any subitems), you should provide the name of a function of two arguments: NoteFile and Window. This function will be applied to these arguments when the menu item is selected.

Returns the item added if successful, NIL otherwise.

Example:

```
(NCP.AddSessionIconMenuItem ' NoteFile '(Foo% Function NC.DoFoo "Performs the
function Foo on this notefile") )
```

(NCP.RemoveSessionIconMenuItem <MenuName> <ItemName>)

Removes the menu item named <ItemName> from the session icon menu specified by <MenuName> (one of 'Card, 'NoteFile, or 'Other). <ItemName> should be only the name of the menu item.

Returns the full item removed if successful, NIL otherwise

Example:

```
(NCP.RemoveSessionIconMenuItem ' NoteFile 'Foo% Function )
```

(NCP.RestoreSessionIconMenu <MenuName>)

Restores the menu specified by <MenuName> (one of 'Card, 'NoteFile, or 'Other) to its initial state. If <MenuName> is NIL, all three menus will be restored.

(NCP.SessionIconWindow)

Returns the session icon window.

(NCP.BringUpSessionIcon <IconPosition>)

Brings up the NoteCards icon at IconPosition. If no IconPosition is given and the icon is already on the screen, it will be flashed. If it is not on the screen, the user will be prompted to place it.

The left button menu on the title bar of a notefile's menu icons: The user may modify this menu using the following functions.

(NCP.AddNoteFileIconMenuItem <Item> <OpenOrClosedOrBoth>)

Adds the menu item <Item> to the NoteFile icon menu. <Item> should be a complete menu item and may contain subitems.. The second part of the menu item (and of any subitems) should be a function of two arguments: NoteFile and Window. Window will be the icon window for the notefile. This function will be applied to these arguments when this menu item is selected. <OpenOrClosedOrBoth> specifies for which type of notefile the new operation will be valid. If <OpenOrClosedOrBoth> is 'Open, the item will appear as an operation for open notefiles. If <OpenOrClosedOrBoth> is 'Closed, the item will appear as an operation for closed notefiles. If <OpenOrClosedOrBoth> is 'Both, the item will be available for both open and closed notefiles.

Returns the full item added if successful, NIL otherwise.

Example:

```
(NCP.AddNoteFileIconMenuItem '(Foo% Function NC.DoFoo "Performs the function
Foo on this notefile") 'Open)
```

(NCP.RemoveNoteFileIconMenuItem <ItemName>)

Removes the menu item named <ItemName> from the NoteFile icon menu.

Returns the full item removed if successful, NIL otherwise.

Example:

```
(NCP.RemoveNoteFileIconMenuItem 'Foo% Function )
```

(NCP.RestoreNoteFileIconMenu)

Restores the NoteFile Icon menu to its initial state.

The middle button menu of a notefile's menu icon: It is now possible for users to add menu items to this menu using the following functions.

(NCP.AddNoteFileIconMiddleButtonItem <Notefile><MenuItems>)

Adds list of menu items <MenuItems> to the middle button menu of the main menu icon corresponding to <Notefile>. These menu items will remain on the menu until the next time the notefile is closed. The second item of each menu item should be an atom that is the name of a function to be called when selected.

(NCP.AddDefaultNoteFileIconMiddleButtonItem <MenuItems>)

Adds list of menu items <MenuItems> to the middle button menu of the main menu icons for all notefiles. These menu items will remain on the menus for the life of the session. The second item of each menu item should be an atom that is the name of a function to be called when selected.

The ShowCards middle button menu on the notefile icon: Users can indicate that certain cards should be added to or deleted from this menu using the following functions.

(NCP.AddSpecialCard <Card>)

Adds <Card> to the list of special cards appearing in the middle button menu on the ShowCards option of the notefile icon (for the notefile containing <Card>).

(NCP.RemoveSpecialCard <Card>)

Removes <Card> from the list of special cards appearing in the middle button menu on the ShowCards option of the notefile icon (for the notefile containing <Card>).

The title bar menus of cards and card types: Users can add menu items to either a specific window containing a card, or to the default menu of a card type.

**(NCP.AddTitleBarMenuItemsToWindow <Win> <Button>
<NewMenuItems><TopOrBottom>)**

Adds the menu items <NewMenuItems> to the title bar menu of <Win>. <Button> should be one of 'Left or 'Middle, corresponding to either the LeftButtonTitleBarMenu or the MiddleButtonTitleBarMenu. <Win> should be the window of a visible notecard. <TopOrBottom> should be one of 'Top or 'Bottom, indicating where in the menu the new items should appear.

**(NCP.AddTitleBarMenuItemsToType <Type> <Button> <NewMenuItems>
<TopOrBottom>)**

Adds the menu items <NewMenuItems> to the title bar menu of all cards of type <Type>. <Button> should be one of 'Left or 'Middle, corresponding to either the LeftButtonTitleBarMenu or the MiddleButtonTitleBarMenu. <TopOrBottom> should be one of 'Top or 'Bottom, indicating where in the menu the new items should appear.

Menu of notefiles: the user may use the list and menu of noticed notefiles maintained by NoteCards.

NCP.NoticedNoteFileNames

Global var containing a list of the currently available notefile names noticed by NoteCards.

**(NCP.NoticedNoteFileNamesMenu <IncludeNewNoteFileFlg> <AllowedOperations>
<InterestedWindow> <Operation>))**

Provides user with a menu of noticed notefile names. <IncludeNewNoteFileFlg> should be non-NIL if new notefiles are allowed. <AllowedOperations> should be one of the atoms: OPEN, CLOSED or NIL for both. <Operation> should be a string or atom containing the name of the operation to be performed on the result and the word NoteFile; e.g. (QUOTE Open% NoteFile). This is used in the prompt for a new notefile name. <InterestedWindow> is the window to receive a prompt window for any messages printed.

(NCP.ForgetNoteFileName <NoteFileOrFileName>)

The notefile is removed from the menu of noticed notefiles. It will not be added to the menu again until explicitly remembered using NCP.RememberNoteFileName regardless of other operations performed on the notefile.

(NCP.RememberNoteFileName <NoteFileOrFileName>)

The notefile is added to the menu of noticed notefiles. This is only necessary if the NoteFile name has been forgotten.

NCP.GrayShade

Global var containing the shade used for shading various menu items in the interface. The default value of NCP.GrayShade is GRAYSHADE. This variable should be changed with the function NCP.SetGrayShade (see below).

(NCP.SetGrayShade <Shade>)

This function should be used to change the value of NCP.GrayShade to <Shade>. In addition to changing NCP.GrayShade, it also resets all cached menus that use NCP.GrayShade. If <Shade> is NIL, NCP.GrayShade will not be changed, but the menus will still be reset. Returns the new value of NCP.GrayShade.

Handy Miscellaneous Functions

(NCP.BringUpNoteCardsIcon <IconPosition>)

Brings up the NoteCards session icon at the position <IconPosition>. If no position is given, will prompt the user to place the icon. If the icon is already on the screen and no position is given, it will be flashed. (equivalent to (NoteCards <IconPosition>).

(NCP.TitleSearch <NoteFile> <keys> <caseSensitiveFlg>)

Returns a list of all cards in <NoteFile> having all of the <keys> (can be atom, string or list) within their titles. Normally case insensitive unless <caseSensitiveFlg> is non-nil.

(NCP.PropSearch <NoteFile> <propOrPair> <propOrPair> ...)

Returns a list of all cards in <NoteFile> such that for every <propOrPair> arg, if it is atomic, then the card contains that property. If it is a list of two elements, then the card must have a property EQ to the first element with value EQ to the second element.

(NCP.WhichCard <x> <y>)

Returns the card currently displayed on the screen whose window contains the position in screen coordinates of <x> if <x> is a POSITION, the position (<x>,<y>) if <x> and <y> are numbers, or the position of the cursor if <x> is NIL. Returns NIL if the coordinates are not in the window of any card. If they are in the window of more than one card, then returns the uppermost. If <x> is a window, then NCP.WhichCard will return the card associated with that window. (The function NCP.WC is an alias for NCP.WhichCard.)

(NCP.WhichNoteFile <x> <y>)

Works just like NCP.WhichCard, returning the notefile corresponding to the indicated window. If the window is for a card, then the card's notefile is returned, if it's for a notefile menu, then that notefile is returned. (The function NCP.WNF is an alias for NCP.WhichNoteFile.)

(NCP.NoteFileIconWindow <NoteFile>)

Returns the main menu icon window, if any, for given <NoteFile>.

(NCP.DisplayedCards <NoteFiles> <CardTypes>)

Returns a list of all cards currently displayed on the screen that are in one of **<NoteFiles>** and are of one of the types in **<CardTypes>**. (Shrunken ones *are* included.) If **<CardTypes>** is NIL, then looks at all card types. If **<NoteFiles>** is NIL, then looks at all notefiles.

**(NCP.SelectCards <instigatingCardOrWindow> <singleCardFlg>
<selectionPredicate> <message> <checkForCancelFlg> <NewCardFlg>)**

Returns a list of those cards selected from the screen. A menu appears in or near **<instigatingCardOrWindow>** displaying **<message>** and having buttons for DONE, UNDO, CANCEL (unless **<singleCardFlg>** is non-nil, in which case there is only a CANCEL button) and NEW CARD (if **<NewCardFlg>** is non-NIL). Card selections must satisfy **<selectionPredicate>** and are made by left buttoning in the title bars of the desired cards. If user hits CANCEL button, then NIL is returned unless **<checkForCancelFlg>** is non-nil, in which case the atom DON'T is returned.

(NCP.DocumentParameters <parametersProplist>)

Returns the old value of the document parameters in the form of a proplist. If **<parametersProplist>** is non-nil then it should be a proplist whose properties are (some of the) valid document parameter names and whose values are permissible values for those parameters. The valid parameters and possible values are as follows:

HeadingsFromFileboxes	NumberedBold, UnnumberedBold, NONE.
TitlesFromNoteCards	Bold, NotBold, NONE.
BuildBackLinks	ToCardsBoxes, ToCards, ToBoxes, NONE.
CopyEmbeddedLinks	ALL, NONE, <listOfLinkLabels>.
ExpandEmbeddedLinks	ALL, NONE, <listOfLinkLabels>.

[See the Notecards user's manual for an explanation of these parameters and how their values affect the document created.]

(NCP.NoteCardsParameters <parametersProplist>)

Returns the old value of the global Notecards parameters in the form of a proplist. If **<parametersProplist>** is non-nil then it should be a proplist whose properties are (some of the) valid document parameter names and whose values are permissible values for those parameters. The valid parameters and possible values are as follows:

DefaultCardType	<legalCardType>
MenuLingersAfterNoteFileClose	T or NIL
ShowNoteFileOnCards	T or NIL
NewNoteFileInitialSize	<positive number (default is 1000)>
ForceFiling	T or NIL
ForceTitles	T or NIL
CloseCardsOffScreen	T or NIL
BringUpCardsAtPreviousPos	T or NIL
MarkersInFileBoxes	T or NIL

AlphabetizedFileBoxChildren	T or NIL
DefaultLinkIconAttachBitmap	T or NIL
DefaultLinkIconShowTitle	T or NIL
DefaultLinkIconShowLinkType	T or NIL
UseDeletedLinkIconIndicators	T or NIL
DelTEditProcessWhenShrinking	T or NIL
ExtraTEditProps	a prop list
EnableBravoToTEditConversion	T or NIL
IncludeCardObjectInShowInfo	T or NIL
LinkDashingInBrowsers	T or NIL
ArrowHeadsInBrowsers	one of the litatoms {AtEndpoint, AtMidpoint, None}
SpecialBrowserSpecs	T or NIL
DefaultFont	a font
LinkIconFont	a font
MenuFont	a font
NoteFileIndicatorFont	a font
Here, <legalCardType> should be an existing Notecard type, i.e., one that appears in the list returned by NCP.CardTypes.	

(NCP.CoerceToInterestedWindow <WinOrCardOrNoteFile>)

Coerces a window, card or notefile into a window which can have a prompt window.

(NCP.PrintMsg <window> <clearFirstFlg> <arg1> <arg2> ...)

Prints a message in the prompt window of <window>. If <window> is NIL, then prints message in the Lisp prompt window. If <clearFirstFlg> is non-nil, then clears the prompt window first. The args are PRIN1'ed one at a time.

(NCP.ClearMsg <window> <closePromptWinFlg> <WaitMsecs>)

Clears the prompt window associated with <window> (or with the main Lisp prompt window if <window> is NIL) and closes it if <closePromptWinFlg> is non-nil. The prompt window will not be cleared before the specified wait has expired.

(NCP.AskUser <Msg> <prompt> <FirstTry> <ClearFirstFlg> <MainWindow> <DontCloseAtEndFlg> <DontClearAtEndFlg> <PROMPTFORWORDFlg>)

This function can be used to ask questions of the user in a window's prompt window. The <Msg> and <prompt> are printed along with <FirstTry> (if non-nil). The value returned is whatever the user types. If <ClearFirstFlg> is non-nil, then the prompt window is cleared first. If <MainWindow> is nil, then the top level prompt window is used. If <DontCloseAtEndFlg> is non-nil, then the prompt window won't be closed after the question is answered and if <DontClearAtEndFlg> is non-nil, then the prompt

window won't be cleared at the end. If <PROMPTFORWORDFlg> is non-nil, then the PROMPTFORWORD typein protocol will be used rather than TTYIN. The former doesn't allow mouse editing of the string typed in. On the other hand, typing automatically overwrites the prompt when PROMPTFORWORD is used.

(NCP.AskYesOrNo <Msg> <prompt> <FirstTry> <ClearFirstFlg> <MainWindow> <DontCloseAtEndFlg> <DontClearAtEndFlg>)

This function can be used to ask yes/no questions of the user in a window's prompt window. The fields are as in NCP.AskUser except that PROMPTFORWORD is always used so there is no Flg for that.

(NCP.CardDates <Card>)

Returns a NOTECARDDATES record structure containing the dates of last modification of each of the four card parts of <Card>. The fields of the record are SUBSTANCEDATE, TITLEDATE, LINKSDATE and PROPLISTDATE.

(NCP.DetermineDisplayRegion <card> <region/position>)

Returns the region that <card> would occupy if displayed. If <region/position> is NIL, then asks the user to position a ghost region to get the position. Also checks previous size of card or default heights and widths for the card type to find the size. If BringUpCardAtPreviousPos is on, then won't require positioning of ghost region even if <region/position> is NIL.

(NCP.SetUpTitleBar <CardWindow> <CardType>)

This function is usually called from the MakeFn for those card types that inherit from a card type that doesn't display in a window, like the List or NoteCard type. It creates and installs left and middle menus using the menu items found in <CardType>'s definition. It also installs a default button event fn on the window.

(NCP.LinkFromLinkIcon <linkIcon>)

If <linkIcon> is an image object for a link icon, then returns the associated link. Note that link icons can be obtained by calling the card type's CollectLinksFn (see the card types mechanism documentation).

(NCP.MakeLinkIcon <link>)

If <link> is a valid link, then creates and returns a new link icon image object containing it.

(NCP.MarkCardDirty <card> <resetFlg>)

Mark <card>'s substance as dirty thus forcing it to be written down at the next save. If <resetFlg> is non-nil, then *unmark* <card> as dirty.

(NCP.MarkAsNotNeedingFiling <card>)

Mark <card> as not needing filing. This is done by setting the card's Don'tRequireFilingFlg prop to T.

(NCP.CoerceToCard <cardIdentifier>)

Return the card object (if any) associated with <cardIdentifier> which can currently be any of: a card object, window or text stream.

(NCP.CollectCards <RootCards> <LinkTypes> <MaxDepth> <FollowCrossFileLinksFlg>)

Starting from <RootCards> and following links having types in <LinkTypes> to a maximum depth of <MaxDepth>, collect and return a list of all cards encountered. A value of NIL for <MaxDepth> causes search to be "infinitely" deep. <LinkTypes> and <FollowCrossFileLinksFlg> are as described in NCP.CardNeighbors.

(NCP.CopyCards <Cards> <DestNoteFileOrFileBox> <RootCards> <QuietFlg> <CopyExternalToLinksMode> <InterestedWindow>)

This copies all cards in <Cards> along with all links among them. If <CopyExternalToLinksMode> is 'COPY, external to-links will also be copied; if it is 'DON'TCOPY, external to-links will not be copied; and if it is NIL, the user will be asked if he/she wishes to copy these links. <DestNoteFileOrFileBox> designates a filebox in which to file the card copies. (If <DestNoteFileOrFileBox> is a notefile, then its Contents box is used.) <RootCards> should be a subset of Cards or NIL. If <RootCards> is NIL, then all the card copies are filed in the destination box, otherwise just those appearing in <RootCards>. If <RootCards> is the atom NONE, then none of the new cards will be filed in the destination filebox. <QuietFlg> cuts out the messages. Note that currently, <Cards> must all live in the same notefile, but this *can* be a different notefile from the destination notefile. <InterestedWindow> is a window used to attach a prompt window for messages.

(NCP.MoveCards <Cards> <DestNoteFileOrFileBox> <RootCards> <QuietFlg> <CopyExternalToLinksMode> <InterestedWindow>)

This moves all cards in <Cards> along with all links among them. If <CopyExternalToLinksMode> is 'COPY, external to-links will also be copied; if it is 'DON'TCOPY, external to-links will not be copied; and if it is NIL, the user will be asked if he/she wishes to copy these links. <DestNoteFileOrFileBox> designates a filebox in which to file the card copies. (If <DestNoteFileOrFileBox> is a notefile, then its Contents box is used.) <RootCards> should be a subset of Cards or NIL. If <RootCards> is NIL, then all the card copies are filed in the destination box, otherwise just those appearing in <RootCards>. If <RootCards> is the atom NONE, then none of the new cards will be filed in the destination filebox. <QuietFlg> cuts out the messages. Note that currently, <Cards> must all live in the same notefile, but this *can* be a different notefile from the destination notefile. <InterestedWindow> is a window used to attach a prompt window for messages.

The following functions allow users to register cards by name for fast access. This is normally done using the card's notefile's system registry. Thus this registering is preserved over notefile closing.

(NCP.RegisterCardByName <Name> <Card> <RegistryCard>)

Stores <Card> in <RegistryCard> hashed under the name <Name>. If <RegistryCard> is nil, then use <Card>'s notefile's system registry.

(NCP.LookupCardByName <Name><NoteFileOrRegistryCard>)

If <NoteFileOrRegistryCard> is a registry card, then recovers the card that was hashed under <Name>. If <NoteFileOrRegistryCard> is a notefile, then use its system registry.

(NCP.UnregisterName <Name><NoteFileOrRegistryCard>)

If <NoteFileOrRegistryCard> is a registry card, then smashes whatever was stored under <Name>. If <NoteFileOrRegistryCard> is a notefile, then use its system registry.

(NCP.ListRegisteredCards <NoteFileOrRegistryCard> <IncludeKeysFlg>)

If <NoteFileOrRegistryCard> is a registry card, then returns the list of cards hashed in it. If <NoteFileOrRegistryCard> is a notefile, then use its system registry. If <IncludeKeysFlg> is non-nil, then return list of cons pairs with car=Name and cdr=Card.

One can associate functionality with the opening and closing of notefiles using two special List cards registered under the names OpenEventsCard and CloseEventsCard. Their substances each consist of a list of lisp s-expressions to be evaluated at notefile open and close time respectively. Thus, to cause a new expression to be evaluated at notefile open time, one cons's (or appends) the expression to the substance of OpenEventsCard (either via DEdit or NCP.CardSubstance). Note that during the evaluation of the expressions, the atom NoteFile is bound to the notefile being opened or closed. When a notefile is closed, the expressions in the CloseEventsCard are evaluated both just before the notefile is closed, and after it is finished being closed. In this case, the atom When is bound to either 'Before or 'After, indicating when the expressions are evaluated with respect to the close operation.

(NCP.GetOpenEventsCard <NoteFile>)

Returns the open events card for <NoteFile> creating a new one if necessary. Thus, this is basically just (NCP.LookupCardByName 'OpenEventsCard <NoteFile>), except that a new List card is created and registered under the name OpenEventsCard if none exists.

(NCP.GetCloseEventsCard <NoteFile>)

Returns the close events card for <NoteFile> creating a new one if necessary. Thus, this is basically just (NCP.LookupCardByName 'CloseEventsCard <NoteFile>), except that a new List card is created and registered under the name CloseEventsCard if none exists.

Sometimes, it's nice to have a place to temporarily hang your hat, so to speak. For this purpose, we provide what are called *UserProps* for both card and notefile objects. These are temporary in that they vanish when the notefile closes.

(NCP.CardUserDataProp <Card><Prop> [<NewValue>])

Returns the value under the <Prop> user data prop for <Card>. If <NewValue> is present, then value under <Card>'s <Prop> is reset. (That is, it works like WINDOWPROP or NCP.CardProp.)

(NCP.NoteFileProp <NoteFile><Prop> [<NewValue>])

Returns the value under the <Prop> user data prop for <NoteFile>. If <NewValue> is present, then value under <NoteFile>'s <Prop> is reset. (That is, it works like WINDOWPROP or NCP.CardProp.)

(NCP.NoteFileAddProp <NoteFile><Prop> [<NewValue>])

Returns the value under the <Prop> user data prop for <NoteFile>. If <NewValue> is present, then it is added to the list under <NoteFile>'s <Prop>. If <NoteFile> has no value under <Prop>, then <NewValue> is placed on that property as a list. (That is, it works like WINDOWADDPROP.)

Error breaks / messages under the programmer's interface: When the programmer's interface detects an error, it calls NCP.ReportError or NCP.ReportWarning depending on the severity of the error. Normally the former causes a break while the latter merely prints a message. However, this behavior can be changed using the global var NCP.ErrorBrkWhenFlg, a litatom whose value should be one of NIL, ALWAYS or NEVER (default is NIL).

(NCP.ReportError <Function><Message>)

Forces a break using BREAK1 unless the value of NCP.ErrorBrkWhenFlg is the litatom NEVER. If so, then simply prints a message and continues.

(NCP.ReportWarning <Function><Message>)

Prints the given warning message <Message> unless the value of NCP.ErrorBrkWhenFlg is the litatom ALWAYS. If so, then forces a break with BREAK1.

There are a few old functions preserved for backward compatibility, but we encourage users to rewrite all their code to use only functions described above. The old functions still available are: NCP.BringUpCard, NCP.ActivateCards, NCP.ActiveCardP, NCP.DeactivateCards, NCP.ValidCard, NCP.GetContentsFileBox, NCP.GetOrphansFileBox, NCP.GetToBeFiledFileBox, NCP.GetLinkSource, NCP.GetLinkDestination, NCP.CreateLinkLabel, NCP.DeleteLinkLabel, NCP.RenameLinkLabel, NCP.GetLinkLabels, NCP.GetUserLinkLabels, NCP.GetReverseLinkLabels, NCP.ValidLinkLabel, NCP.AddLeftButtonTitleBarMenuItems, NCP.AddMiddleButtonTitleBarMenuItems.

[This page intentionally left blank]

APPENDIX C - NOTEFILE INSPECTOR

The main purpose of the notefile inspector is to repair notefiles by rebuilding their links. However, the inspector is generally useful for checking the health of notefiles, deleting cards and backing up cards (or more precisely, card parts) to previous versions. Thus, you may want to use the inspector even if your notefile is healthy and doesn't need its links rebuilt.

The notefile inspector has three separate phases:

- Reading the notefile's data area searching for healthy card parts
- Allowing the user to make modifications
- Rebuilding the links

The process can be aborted after Phase 1 or 2 if desired. This document describes each of the three phases in turn and concludes with tips, strategies and pitfalls to watch for.

Using the notefile inspector requires some knowledge of the innards of notefile organization. We recommend that you read Appendix A, Notefile Concepts, before continuing.

Note: The information in this appendix was copied whole from the documentation for a prerelease of NoteCards and has not yet been updated for this release. As such, it may be inaccurate in places.

Running the Notefile Inspector: Phase 1: Scouring the Data Area

To start the inspector on a notefile, first be sure that the notefile is not open. Then select the item **Inspect&Repair** from one of the NoteFile Ops menus (i.e. from the notefile Banner, the MenuBox icon or the notefile FileBrowser). There is one option available at this level by "pulling to the side" called **ReadSubstances**. This ensures that substances of all cards pronounced valid by the inspector are readable. If this option is not invoked, then a check is still run on the length of the substance, but not on its contents. Unfortunately, the ReadSubstances option requires MUCH more work by Phase 1. We recommend that you only use this option if Phase 3 (link rebuilding) breaks with some error like "Bad Piece Tbl" from TEdit. In that case, up-arrow out of the break and start the Inspect&Repair process over again, this time using the slower but more comprehensive ReadSubstances option.

Selecting **Inspect&Repair** will invoke phase 1 of the inspector, wherein the data area of the notefile is scoured for valid card parts. A record of all such parts is kept and statistics printed out at the end. You'll be asked to position the window in which those statistics as well as later inspector communications will be printed. (Note that closing this window will abort the Inspect&Repair process if you confirm.) You can monitor the progress of Phase 1 by watching the prompt window. It will be printing messages like "Processing byte xxxxx of yyyy."

When phase 1 has completed and you've positioned the interaction window, statistics on your notefile will be provided. You'll be told the total number of cards and the number of deleted ones. If all seems well with the world, the next line will read "All non-deleted cards look okay." If not, there will be various messages outlining the problems. (See Figure C-1.)

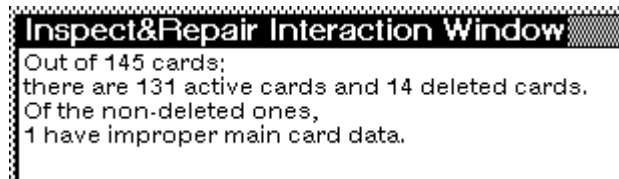


Figure C-1: Snapshot of a Sample Interaction Window

Normally, the messages describing problems are of the form shown in Figure C-1, namely a count of the number of cards broken in a given way. Some messages are a bit trickier. If you have fileboxes with bad substances (i.e., main card data), then you're told that if you don't wish to delete or back up such fileboxes to previous versions, then Phase 3 will rebuild them. If, however, the filebox is one of the top level boxes (Contents, Orphans, ToBeFiled), then deletion is not an option. (See Section 3.)

If there are cards having user-defined types whose type definition code has not been loaded, then you'll get a message to that effect, something like "<n> cards have unknown card types (FOO BAR)." At this point you should load the lisp files containing the definitions of the unknown card types. If not, then these cards will show up with bad substance in Phase 2. If you do load the appropriate files, then run **Recheck Bad Cards** to get the bad cards list recomputed before inspecting any cards.

If you have a card for which no substance versions could be read, then you'll also get unknown card type messages for it (reading something like "<n> cards have unknown card types (NIL)"). This is because the inspector couldn't find a card type on the notefile for that card.

A menu of options appears attached to the upper right corner of the interaction window. (See Figure C-2.) The particular options you get in that menu depend on the state of your notefile and are described below. The last three options appear in all cases. The other two may or may not be present in the menu you get. In any case, you should select one of the options before attempting any other NoteCards-related work.



Figure C-2: Phase 1 Options Menu

Continue Repair

This option is only available if the notefile is in fairly good health (i.e., okay except for fileboxes to rebuild or global links to rebuild - see Section 3). Selecting it causes Inspect&Repair to move to Phase 3 and rebuild your notefile's links.

End Inspect&Repair

This option is only available if it seems that you don't need to continue to the link rebuilding phase. You will not get this option if you've deleted any cards, or generally if there are problems with the notefile. Choosing this option causes the Inspect&Repair process to end gracefully (via a normal checkpoint and close notefile), thus skipping Phase 3 of rebuilding links.

Abort

Choosing this option aborts the Inspect&Repair process entirely, throwing away any changes you might have made (such as card deletions or back ups.) It requires your confirmation in the prompt window.

Recheck Bad Cards

Recompute the bad cards list by running the last part of Phase 1 again. This is useful if you've loaded files containing definitions of previously undefined card types.

Inspect Cards

This brings up a menu of titles of active cards with which you can inspect, delete, or back up particular cards. There is a "pull-across" menu item called **Include Deleted Cards**, which if selected will include card titles for deleted cards as well as active ones. Using this option, one can undelete deleted cards and restoring them to some previous version.

Running the Notefile Inspector: Phase 2: Your Chance to Tinker

After selecting **Inspect Cards** in the interaction window's attached menu, a menu containing numbered titles of notecards pops up and is attached to the interaction window's lower left corner. It contains the first 10 characters of titles for all active cards and possibly deleted cards as well if you selected the submenu item **Include Deleted Cards** described above. The menu can hold some 100 card titles. If your notefile has more than that, then the menu is composed of several pages each containing around 100 cards. Rapid switching between pages is possible. Figure C-3 below shows a sample card inspector menu.

Attached to the upper right corner of the cards inspector menu is a menu containing at least the three options: **Abort**, **Done** and **Search**. If the menu has multiple pages (there are more than 100 active cards in the notefile), then the attached menu will also include the items **Next Page**, **Previous Page**, and **First Page**. Selecting these causes the current menu to be exchanged for either the next menu, previous menu, or first menu, respectively.

Clicking **Abort** causes the entire Inspect&Repair process to quit, throwing away any changes you've made. It requires your confirmation in the prompt window. (This is equivalent to choosing **Abort** from the inspector window as described in the previous section.)

Choosing **Done** from this attached menu indicates that you're done tinkering with card parts and wish to return to the main interaction window. This causes the card inspector menu to close and the Phase 1 process outlined in the previous section to be performed again (leaving you looking at something like Figure C-1). Note, however, that the scanning of the data area is not repeated (it takes way too long). Rather, your changes are made to the in-core index array and the statistics on bad cards are recomputed. This cycle of compute statistics (Phase 1) followed by inspect and tinker (Phase 2) can be repeated as often as desired. Eventually, you must either abort, end the Inspect&Repair via **End Inspect&Repair**, or continue to Phase 3 via **Continue Repair**.

Choosing **Search** from the attached menu allows you to find those cards having titles matching a given string. This works much like the Search card in Notecards. That is,

you specify a string and the list of cards whose titles contain that string is printed out. This is handy, for example, if you want to undelete a card, but only know its title. Note that the search is case *insensitive*.

Card Inspector Menu

In the card inspector menu, those titles corresponding to deleted cards have a line drawn through them. Those having some sort of problem appear shaded. In addition, an upper-case letter suffix is attached to such entries indicating the problem. For example, in the card inspector menu shown in Figure C-3, the shaded menu item "10: BNF for Li|S" indicates that the card with title beginning "BNF for Li" has bad substance. The letter codes are S, L, P, and T indicating bad substance, links, property list, and title, respectively. If such a letter code appears in lower case, then the meaning is that the current version of that card part is beyond the last checkpoint pointer. (There may have been a crash, for example, thus preventing the notefile from closing normally.)

Figures C-3 and C-4 show both pages of a two page card inspector including deleted cards. Sometimes deleted cards (e.g. "110: NIL|LSPT" in Figure 4) show up shaded with LSPT suffixes. This is because they were deleted before any data about the card was written to the notefile. Thus there are no valid card parts to back up to for those cards. But often, deleted cards can be backed up to previous versions. (See Figure 6 and discussion below.)

Card inspector: Page 1 of 2				Abort Done Search Previous Page Next Page First Page
1: Contents	2: Orphans	3: To Be File	4: Link Label	
5: Registry	6: Papers	7: Value Cell	8: Value cell	
9: Introducti	10: BNF for Li S	11: BNF for Co	12: Format of	
13: Recomputin	14: Example fr	15: Implicatio	16: Extensions	
17: Formula ex	18: Collect ex	19: Outline	20: Value cell	
21: People	22: Schatz, Br	23: Sharma	24: Dean, Jeff	
25: Kasif, Sim	26: Krovetz, B	27: Liles, Bil	28: Feuerman,	
29: Cook, Lesl	30: Kondo, Jin	31: Schwartz,	32: Ward, Sand	
33: Gregory, R	34: Schwartz,	35: Delisle, N	36: Chapman, G	
37: Hamlet, Di	38: Ellis, Ski	39: Halasz, Fr	40: Brennan, S	
41: Moore, Lee	42: VanLehn, K	43: Carnese, D	44: Notecards	
45: NC in Loop	46: Review of	47: Pros and C	48: Introducti	
49: Discussion	50: Paper outl	51: Document f	52: Reggia, Ji	
53: Wohltman,	54: junk	55: NoteCards	56: Applicatio	
57: Applicatio	58: Introducti	59: Random not	60: NCAPPLICAT	
61: Mail box	62: Question o	63: The progra	64: ARI Examl	
65: Second exa	66: Summary of	67: Notes from	68: Mont-Reyna	
69: Kim, Scott	70: Weiser, Ma	71: Places	72: REI	
73: Frescoe's	74: Higgins, R	75: Dieter's	76: Pitts, Wei	
77: Leavitt, M	78: Marshall,	79: Kimball's	80: Deerwester	
81: CPSR	82: Moran, Tom	83: Nelson, Te	84: Informatio	
85: Mail	86: Loops	87: Colab	88: Koncepts C	
89: Wijsman, N	90: Todd, John	91: Draft	92: Notes on K	
93: Stats on K	94: Cards	95: Links	96: Notefile	
97: Notes from	98: Thoughts o	99: Document f	100: Notes from	

Figure C-3: Card Inspector Menu: Page 1

Card inspector: Page 2 of 2		Abort
102; Tailorabil	103; Global par	Done
104; Interlisp	105; Card regis	Search
100; PI-based f	107; Event hook	Previous Page
108; Template c	109; Interlisp	Next Page
110; Programmer	111; Types mech	First Page
112; Card types	113; Tailorabil	
114; foo	110; foo	
117; asdf	110; WHL-GPT	
119; Table of P	120; Table of c	
121; Table of c	122; Support fo	
123; Outline	124; Techniques	
125; Document f	126; PI-based f	
127; Event hook	128; Document f	
129; Thoughts	130; Talks	
131; Talk on NC	132; Types hier	
134; Title slid	135; Levels of	
136; Global par	138; PI sketch	
139; Use of PI	140; Event hook	
141; Types mech	142; Miscellane	
143; IDE exampl	144; Kinds of c	
145; Estrin, De		

Figure C-4: Card Inspector Menu: Page 2

At the top of the first page of any notefile's card inspector, there are entries for the top level boxes and special cards. These are labeled "1: Contents", "2: Orphans", "3: To Be File", "4: Link Label", and "5: Registry".

Note that if the inspector shows no entry in the card inspector menu for some card, it is because that card has been deleted. If you've asked to show deleted cards and it still doesn't appear, it's because the notefile has been compacted since that card was deleted.

If you button card title entry in the card inspector menu, then a popup menu allows up to two choices **Inspect** and/or **Delete**. If the card is currently deleted (has a line drawn through it), then the Delete option is replaced by **Undelete**. Certain cards cannot be deleted and thus their popup menus only contain the Inspect option. These are the 5 special cards mentioned above.

Choosing Delete or Undelete from this popup menu causes the card to be deleted or undeleted, respectively and the line through the menu item either drawn or undrawn. Note, however, that this action (and all others) can be undone by choosing ABORT from either the card inspector menu or the interaction window menu.

Choosing Inspect from the popup menu for a card entry brings up a card parts inspector for that card.

Card Parts Inspector

Figures C-5 and C-6 below show examples of card parts inspectors. A card parts inspector is composed of four attached menus arranged vertically and one attached operations menu atop the stack.

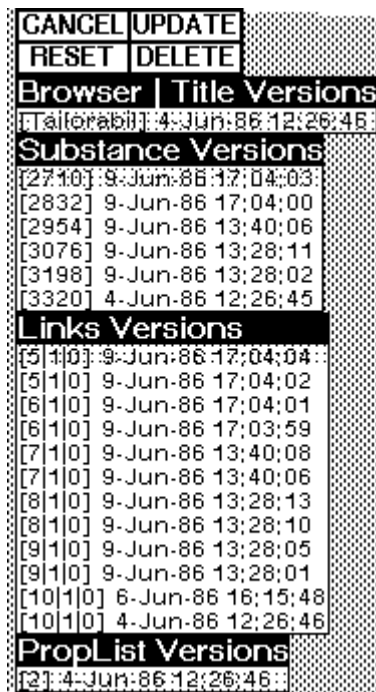


Figure C-5: Card Parts Inspector for a Browser Card

In Figure C-5, the four menus contain entries for every valid version of card parts for the browser card with title "Tailorability". The top menu is for versions of titles and below that are menus for versions of the card's substance, links, and prop list. For example, the Substance submenu contains entries for 6 versions of the substance of this card. The current version of each card part is shaded. Each menu entry gives the date that that version was written. (Very old notefiles may show "NO DATE AVAILABLE" in place of a real date.)

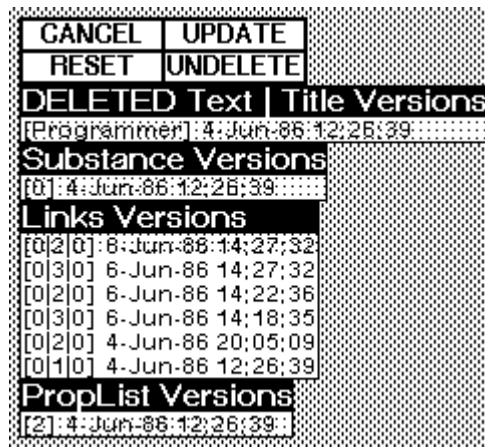


Figure C-6: Card Parts Inspector for a Deleted Card

Figure C-6 shows a card parts inspector for a deleted text card and thus allows the UNDELETE option. By clicking UNDELETE, this card can be backed up to its last version.

If the current version of the card part is bad, then the menu entry will be a string so indicating, for example, "BADSUBSTANCE."

The title of the top menu includes the card's type. In addition, each menu item contains some information, in square brackets, before the date. In the title versions menu, this

information is the first few characters of the title. In the substance versions menu, it is the number of bytes in the substance. In the links versions menu, it is a triple of numbers giving the number of to links, from links, and global links for this card. Finally, the proplist versions menu includes the number of entries on the property list for this card (i.e. twice the number of attribute-value pairs).

Atop the stack of menus is an attached menu of operations, described below.

CANCEL

This cancels this card parts inspector, throwing away any changes made.

UPDATE

This closes the card parts inspector, incorporating any changes (backing up to previous versions of card parts) you might have done.

DELETE

This option closes the card parts inspector and deletes the card.

UNDELETE

For cards that have been deleted, this option appears instead of DELETE. Choosing it causes the card to be undeleted.

RESET

This causes the selections in the submenus to be restored to the values they had when the card parts inspector was first brought up. (Equivalent to doing CANCEL and then inspecting this card again from the card inspector menu.)

Note that cards that can't be deleted (like the top level fileboxes) don't have the DELETE option on their card parts inspector.

Buttoning an entry in a submenu of a card parts inspector pops up a short menu unless the entry is for a bad version (e.g. "BADSUBSTANCE"). This menu contains at least the entry **Inspect** and possibly **Change Selection**, if the selected entry is not the same as the current one (i.e. not shaded).

Choosing Inspect allows further inspection of the details of the selected card part version. (See Figure C-7 below.) For example, inspecting a title version brings up the Interlisp inspector on a record containing the title, date and card object. Similarly, inspecting a links or prop list version brings up the Interlisp inspector on a record containing the lists of links (i.e. to links, from links and global links) or the prop list. Note that changing values in these Interlisp inspectors has no effect on the notefile and is ignored.

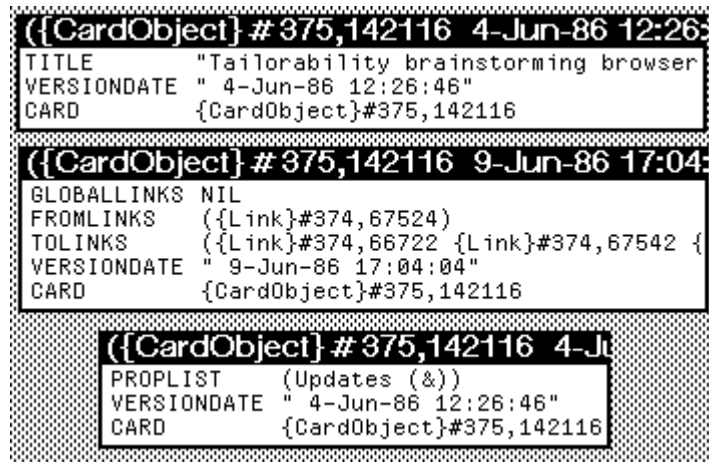


Figure C-7: Card Part Version Inspectors for Title, Links and Prop List

All substance versions for card types inheriting from Text, Sketch and Graph can currently be inspected. (This includes most cards except those having user-defined substance types, like the NCFfile card.) Inspecting a card's substance version will bring up a window showing a copy of the substance. (Note that changes to this copy have no affect on the notefile.) Any links in the substance of the card will show up as bracketed strings describing the link.

Choosing **Change Selection** from the card part version popup menu causes the current selection to be changed to the selected one. This changes a card part's current version to another legal one. (This change can be undone by resetting or canceling the card parts inspector as well as by later aborting the card inspector or interaction window.)

Running the Notefile Inspector:

Phase 3: Rebuilding Your Links

To complete the Inspect&Repair process, select the **Continue Repair** option from the interaction window menu. This invokes Phase 3, the links rebuilder. Normally, this simply rebuilds links from card substances. In certain circumstances, it may do extra work as well. If your link types list is bad, and you didn't back it up to a previous version, then Phase 3 will rebuild it. If there are fileboxes with bad substances that you haven't either deleted or backed up to previous versions, then Phase 3 will rebuild them. Finally, if there are cards with bad links that you haven't backed up or deleted, then Phase 3 will rebuild those links as well. (It rebuilds ALL to links and from links anyway. For those cards, it will rebuild global links as well.)

First the links rebuilder removes all the to and from links for every card. Then it reads the substances for each card and recreates to links and from links by looking at the links found inside the link icons in the card's substance.

The link rebuilder is also able to rebuild bad filebox substances. It does this by looking for all cards in the notefile with from links from the bad box and creating a new substance for the box containing only links to those cards. This process loses any text that the box might have contained as well as scrapping the original ordering of links. Nonetheless, in some cases this may be preferred to backing up the substance to a previous version or to deleting the box altogether.

The links rebuilder can rebuild the notefile's list of link types in a similar manner. That is, it records the set of link types seen on valid links and replaces the old links

types with the new set. Note that this throws away any link types for which there are no links in the notefile.

Finally, the links rebuilder can rebuild bad global links for a card. It does this by looking for any cards with from links from the bad card that are global. This assumes that the card at the destination end has good links. Thus, if the cards at both ends of a global link have unreadable links, then there is no way to recover that link.

Note that the links rebuilder automatically files any unfiled cards in the ToBeFiled box. A message is printed in that case.

Tips and Hints for Using Inspect&Repair

This section contains a list of strategies and tips for using Inspect&Repair. For the most part, they are ordered from the useful and obvious to the esoteric. Several of these are implicit in the first four sections of the document, but are repeated here for emphasis and completeness.

When in Doubt, Abort!

All your changes will be lost, but then if you're uncertain about what's happened this is the safe course. Often, in fact, you may simply want to check the health of your notefile and abort without tinkering.

Fixing versus Tinkering

There are two main ways to use the inspector, either for fixing a broken notefile, or tinkering with a healthy one. The latter case occurs when you wish to recover some card that was inadvertently deleted. Or back up a card that was accidentally changed to its original version.

Inspect&Repair After a Crash

If you try to open a notefile and you get the infamous "Work was done since last checkpoint..." message, then you're given the chance to run Inspect&Repair on the notefile. This is a good idea whenever you wish to recover that post-checkpoint work. The inspector will integrate the changes you made since that checkpoint. (You do have to continue through Phase 3, however.)

Compacting

It used to be the case that old versions of card parts were in the notefile but inaccessible. Thus, there was little reason not to compact a notefile often. Now there is a tradeoff between the need to save space by compacting versus the need to be able to back up using Inspect&Repair. Probably the safest course is to keep a backed up copy of the pre-compacted notefile around until you have confidence that the compacted one is healthy and that you have no need for previous versions of any of its cards.

Inspect&Repair Closes the Notefile Before Starting

This means that if you are working in your notefile and notice a card you'd like to inspect a previous version of, you must record the card's title and close the notefile. Then, run Inspect&Repair, find the card's entry in the inspector menu using the Search facility, and tinker with it as desired.

Fixing Enough Problems to Allow Phase 3 to Run

You can't run Phase 3 unless Inspect&Repair thinks your notefile is above a certain threshold of health. There are certain problems it can handle (e.g. bad filebox substances, see Section 3), and others that it can't (e.g. bad substance for non-filebox). You have to decide either to fix these sorts of problems yourself in Phase 2, let Phase 3 attempt to rebuild them, or just abort the whole thing (always an option).

Sometimes these decisions can be tricky. For example, suppose a filebox's substance is bad. Call it BadBox. Should you (a) delete BadBox altogether, (b) back its substance up to a previous version, or (c) allow Phase 3 to rebuild it by looking for from links in other cards from BadBox? Option (c) may not be advisable if there was important text in BadBox or if the order of cards in BadBox was important. On the other hand, option (b) may be of little use if the last good version is too out of date (or if there is no good version at all).

APPENDIX D - INITIALIZATION FILES

Once your system administrator has installed the NoteCards software on your Sun, you can customize your NoteCards environment. The initialization file allows you to specialize your NoteCards environment to the idiosyncracies of your individual site. This file is automatically loaded when you start a fresh sysout.

Site Initialization File

When NoteCards starts, it reads in a site initialization file. This file sets things like the pointer to fonts, printers, and other site-specific parameters.

Locating the Initialization File

When running under SunOS, NoteCards looks for a site initialization file in a number of locations.

LDEINIT If the UNIX environment variable `LDEINIT` is set to a complete NoteCards file name, NoteCards looks there first for the site initialization file:

```
prompt% setenv LDEINIT /users/smith/site-init.lisp
```

- or -

```
prompt% setenv LDEINIT /users/smith/nc-init
```

/usr/local/lde/site-init.lisp If `LDEINIT` is not set or there is no file with the name given, NoteCards looks for a site initialization file called `/usr/local/lde/site-init.lisp`. The distribution tape contains a standard site initialization file in the NoteCards lisp directory `/usr/local/lde/lisp/init.NoteCards` which is linked to `/usr/local/lde/site-init.lisp`. You or your system administrator should customize this file for your site. The comments below and in the standard `site-init.lisp` describe the parameters it sets and gives some guidelines for customizing it to your local conditions.

{DSK}INIT.LISP NoteCards looks for a site initialization file on your NoteCards home directory (`{DSK}`). Chapter 4 describes the `{DSK}` device.

Finally, to load an initialization file after you have been working in a sysout, FileBrowse the directory the initialization file is in, select the file with the left mouse button, and select the **Load** option. The file will load and reset all of the initialization variables.

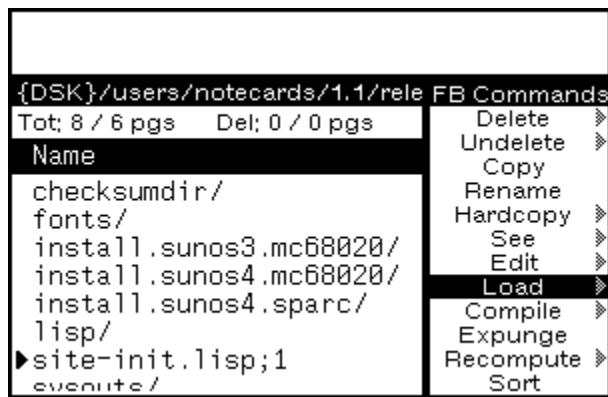


Figure D-1. A FileBrowser loading an initialization file.

Initialization File Structure

The structure of each command in the initialization file is as follows:

```
(SETQ variable-name value)
```

Each command must appear between parentheses. There must always be an even number of parentheses. Every open parenthesis must have a corresponding close parenthesis. SETQ is the assignment command. The variable name is always a single word, and the value is usually either a single word preceded by a single quote mark (') or a list of words enclosed in parentheses preceded by a single quote mark.

The initialization file must always start with two specific lines. The first line must be a comment line, indicated by starting the line with a semicolon (;). The second line must contain the command (CL:IN-PACKAGE "INTERLISP").

Site Variables

The following NoteCards variables should be set in your site initialization file.

LISPUSERSDIRECTORIES

The list of paths to search for library files. Every path in this list should also be in DIRECTORIES.

```
(SETQ LISPUSERSDIRECTORIES
  ' (" {DSK}/usr/local/lde/lispusers/"
    " {DSK}/usr/local/lde/lisplibrary/"))
```

Note that each directory name is a string enclosed in double quotes ("). Unless you are a developer who is modifying NoteCards, there will probably never be any reason to change these values. You only need to change the value for this variable if you move the files in these directories to some other location, such as a central fileserver.

DIRECTORIES

The list of paths to search for files that are not found in the current NoteCards connected directory. The current NoteCards connected directory is distinct from the current UNIX connected directory, and NoteCards will not necessarily search for files in the UNIX connected directory.

```
(SETQ DIRECTORIES (COPYALL LISPUSERSDIRECTORIES))
```

Note that there is *no* quote mark (') before the parenthesis in front of COPYALL. If you set DIRECTORIES in this fashion, this command must follow the one which sets LISPUSERSDIRECTORIES.

You can also simply copy the directories from LISPUSERSDIRECTORIES, e.g.

```
(SETQ DIRECTORIES
  ' (" {DSK}/usr/local/lde/lispusers/"
    "{DSK}/usr/local/lde/lisplibrary/"))
```

Unless you are a developer who is modifying NoteCards, there will probably never be any reason to change these values.

DISPLAYFONTDIRECTORIES

A list of directories to search when the system is looking for display fonts. The site initialization file should set DISPLAYFONTDIRECTORIES to a list of of path names where each path name is represented as a string in double-quotes ("), e.g.,

```
(SETQ DISPLAYFONTDIRECTORIES
  ' (" {DSK}/usr/local/lde/fonts/display/presentation/"
    "{DSK}/usr/local/lde/fonts/display/publishing/"
    "{DSK}/usr/local/lde/fonts/display/printwheel/"
    "{DSK}/usr/local/lde/fonts/display/miscellaneous/"
    "{DSK}/usr/local/lde/fonts/display/jis1/"
    "{DSK}/usr/local/lde/fonts/display/jis2/"
    "{DSK}/usr/local/lde/fonts/display/chinese/"))
```

If you remove the Chinese and Japanese fonts from the font directories it is not necessary to edit this variable to reflect the change.

You only need to change the value for this variable if you move the fonts to some other location, such as a central fileserver. If you do change the value for this variable, note that each location is represented as a double quoted string and that each path name must end with a slash (/).

INTERPRESSFONTDIRECTORIES

A list of directories for the system to search for Interpress font widths files. The site initialization file should set INTERPRESSFONTDIRECTORIES to a list of of path names where each path name is represented as a string in double-quotes ("), e.g.,

```
(SETQ INTERPRESSFONTDIRECTORIES
  ' (" {DSK}/usr/local/lde/fonts/interpress/presentation/"
    "{DSK}/usr/local/lde/fonts/interpress/publishing/"
    "{DSK}/usr/local/lde/fonts/interpress/printwheel/"
    "{DSK}/usr/local/lde/fonts/interpress/miscellaneous/"
    "{DSK}/usr/local/lde/fonts/interpress/jis1/"
    "{DSK}/usr/local/lde/fonts/interpress/jis2/"
    "{DSK}/usr/local/lde/fonts/interpress/chinese/"))
```

If you remove the Chinese and Japanese fonts from the interpress font directories it is not necessary to edit this variable to reflect the change.

You only need to change the value for this variable if you move the interpress fonts to some other location, such as a central filesaver. If you do change the value for this variable, note that each location is represented as a double quoted string and that the lowest subdirectory name in each path must end with a slash (/).

POSTSCRIPTFONTDIRECTORIES

The list containing the name of the PostScript font width files, for PostScript printers.

```
(SETQ POSTSCRIPTFONTDIRECTORIES
  ' (" {DSK} /usr/local/lde/fonts/postscript/"))
```

You only need to change the value for this variable if you move the PostScript fonts to some other location, such as a central filesaver. If you do change the value for this variable, note that the location is represented as a double quoted string and that the lowest subdirectory name in the path must end with a slash (/).

USERGREETFILES

The list of places to search for personal initialization files. If not set in the site initialization file, no personal initialization file is used. The list should be similar to the following

```
(SETQ USERGREETFILES
  ' ( (" {DSK} ~/lde/INIT.LISP") (" {DSK} ~/INIT.LISP") ) )
```

This will search for the file `INIT.LISP` in your home directory and in the subdirectory `lde` immediately under your home directory. The file `INIT.LISP` can be renamed to what ever you want it to be, but it has to be a name which is used by all users of the NoteCards system.

Unless you are a developer who is modifying NoteCards, there will probably never be any reason to change these values or use user greet files.

DEFAULTOSTYPE

Specifies the default operating system type for file servers.

```
(SETQ DEFAULTOSTYPE 'UNIX)
```

Unless you are using other than Sun NFS (Network File System) servers there is no reason to change the value of this variable.

DEFAULTPRINTINGHOST

A list of names of default printers.

```
(SETQ DEFAULTPRINTINGHOST NIL)
```

`NIL` is the default setting and signifies that there are no default printers specified. This variable can also be set from the **Set Default Printer** background menu option, (see Chapter 16, Printing).

The following example shows how to set DEFAULTPRINTINGHOST to a mixed list of PostScript and Interpress printers.

```
(SETQ DEFAULTPRINTINGHOST
  ' (MAUI |Tremor:mv:envos|))
```

- or -

```
(SETQ DEFAULTPRINTINGHOST
  ' (MAUI "Tremor:mv:envos"))
```

MAUI is the name of a PostScript printer, |Tremor:mv:envos| the name of an interpress printer. Each interpress printer name must be surrounded by vertical bars (|) or double quotes (").

DEFAULTPRINTERTYPE

The default printer type for all the available printers.

```
(SETQ DEFAULTPRINTERTYPE NIL)
```

Possible values are 'POSTSCRIPT, 'INTERPRESS, and NIL meaning unspecified. Note that 'POSTSCRIPT and 'INTERPRESS are all preceded by single quote marks ('), but that NIL is not.

If all your printers are PostScript printers you would make the following setting.

```
(SETQ DEFAULTPRINTERTYPE 'POSTSCRIPT)
```

If all your printers are not of the same type set DEFAULTPRINTERTYPE to NIL, and add the following command for each printer.

```
(PUTPROP 'Printer-Name 'PRINTERTYPE 'Type)
```

Where 'Printer-Name is the name of the printer and 'Type is one of 'POSTSCRIPT, 'INTERPRESS, or 'PRESS, for example.

```
(PUTPROP 'MAUI 'PRINTERTYPE 'POSTSCRIPT)
```

```
(PUTPROP '|Tremor:mv:envos| 'PRINTERTYPE 'INTERPRESS)
```

IDLE.PROFILE

The lines below set the IDLE.PROFILE such that any user with a valid UNIX login is allowed to exit idle mode.

```
(LISTPUT IDLE.PROFILE 'AUTHENTICATE 'UNIX)
```

The value of AUTHENTICATE determines what mechanism is used to check passwords. If 'UNIX, it checks your password with SunOS. If T, it uses the NS authentication protocol (this requires the presence of an NS Authentication server accessible via the network). If NIL, the password is not checked at all--any password is accepted. The latter is only useful if ALLOWED.LOGINS contains *.

```
(LISTPUT IDLE.PROFILE 'ALLOWED.LOGINS '(*))
```

The value of ALLOWED.LOGINS determines who is allowed to exit idle mode. If the value is NIL no login is required at all to exit idle mode. If the value is '(*), the system requires a login but lets anyone exit idle mode.

```
(LISTPUT IDLE.PROFILE 'SAVEVM NIL)
```

This line sets the `IDLE.PROFILE` such that NoteCards does not save the virtual memory file to disk when it enters idle mode. To have NoteCards automatically save the virtual memory file to disk when it enters idle mode, replace `NIL` in the line above with `T`.

`\\BeginDST`

The day of the year on or before which Daylight Savings Time takes effect (i.e., the Sunday on or immediately preceding this day). Must be set to 98 in the USA if NoteCards is to perform time computations correctly. (Note: This number is subject to future Congressional legislation.) If you are in a region where Daylight Savings Time is not observed, use the value 367.

`(SETQ |\\BeginDST| 98)`

`\\EndDST`

The day of the year on or before which Daylight Savings Time ends. Must be set to 305 in the USA.

`(SETQ |\\EndDST| 305)`

`XCL:*LONG-SITE-NAME*`

A long version of your company's name, for example, "Any Corporation."

`(SETQ XCL:*LONG-SITE-NAME* "Any Corporation")`

`XCL:*SHORT-SITE-NAME*`

A short version of your Company's name, for example, "Any."

`(SETQ XCL:*SHORT-SITE-NAME* "Any")`

[This page intentionally left blank]

APPENDIX E. CHECKSUM CONTROL

If you encounter inexplicable problems shortly after you install NoteCards, they may be due to files being corrupted — the release tape may have been damaged, errors may have occurred while the tape was being read, etc. If you have unexplained problems, we recommend that you verify the checksums of your installed files.

The script generates checksum files named *filename*.check and compares them to the released *filename*.sum residing in the /checksumdir subdirectory.

The checksum script reports inconsistent files, the correct checksum values for the files, and an error message. The checksum of individual files can be generated with the UNIX command `sum filename`.

Commands

`ldechecksum [-cg] [ncdir [dir | dirgroup]]` (Command)

`-c` Generates checksums for your installed files and compares them with correct values. This is the default action.

`-g` Generates checksums for the files specified.

ncdir Name of the NoteCards installation directory. Usually it is /usr/local/lde.

dir Any specific directory residing under ncdir. Only relative pathnames with respect to ncdir are accepted.

dirgroup The directory group. Valid directory groups are listed below.

`system` includes lisp, sysouts, install.sunos3.mc68020, install.sunos4.mc68020, install.sunos4.sparc

`fonts` includes the display, InterPress, Press and PostScript directories

`all` includes all of the above

Output

As it begins checking each directory, the script prints a message in the following form:

Checking directory: /usr/local/lde/subdir

Error and warning messages may be in one of these two forms:

< E > 36610 6973 NoteCards.sysout

Indicates that file `NoteCards.sysout` is erroneous or does not reside in the directory. The correct checksum of 36610, together with the size (6973 Kbytes) of the file are shown.

```
< W > /usr/local/lde/fonts/display/chinese:  
Directory not installed
```

Indicates that file Chinese fonts were not installed or were removed after NoteCards was installed.

Examples

```
prompt% ldechecksum /usr/local/lde
```

All files in the installed NoteCards directories in /usr/local/lde are checked.

```
prompt% ldechecksum /usr/local/somedir/lde system
```

This example checks all files in:

```
/usr/local/somedir/lde/install.sunos3.mc68020  
/usr/local/somedir/lde/install.sunos4.mc68020  
/usr/local/somedir/lde/install.sunos4.sparc  
/usr/local/somedir/lde/sysouts  
/usr/local/somedir/lde/lisp
```

```
prompt% cd/usr/local/lde
```

```
prompt% ldechecksum -c . fonts/display
```

This example checks only the display font directories. The period (.) is used because you are positioned under the current NoteCards installation directory.

INDEX

A

Abort (*Option*) 106,127
Abort All (*Option*) 128
Abort and Abort NoteFile (*Option*) 128
Abort current process 31
Aborting Operations 185
Add Border (*Option*) 102
Add Global Link (*Option*) 90
Add Label (*Option*) 97
Add Link (*Option*) 96
Add New Property (*Option*) 86
Add New Regional Time Zone (*Option*) 166
Add Node (*Option*) 97
Add to which directory group? (*Menu*) 176
Allowed Logins (*Menu*) 56
Allowed Logins (*Option*) 56
Alphabetized FileBox Children (*Parameter*) 141
Analog clock 167
 changing 168
 known problems 170
 starting 168
 stopping 168
Any Login (*Option*) 57
Arrow Heads In Browsers (*Parameter*) 142
Arrowheads (*Option*) 114
Assign Title (*Option*) 88
Asterisk 23
Attach Sketch Menu (*Parameter*) 142
Authenticate (*Option*) 57

B

Back links 109,111
Background (*Menu*) 53
Backslash 20
Banner, noteFile 79
\\BeginDST (*Variable*) 240
Bitmap
 operations 100
 texture editor 102
Bitmap Editor (*Menu*) 104
Bold (*Option*) 110
Bouncing Box (*Option*) 56
Bouncing Username (*Option*) 56
Boxes only (*Option*) 111
Break windows 32,186
Bring Up Card/Box (*Option*) 75
Bring Up Cards At Previous Pos (*Parameter*) 141
Browser cards 84,112
 parameters 141
Browser Specs (*Option*) 114
Browser Specs? (*Menu*) 113
BrowserOverviewWin (*Option*) 116
BuildBacklinks (*Option*) 111
Bury (*Option*) 49

C

C-Shell 20
Can't Convert a POSTSCRIPT for a NIL Printer
 (*Error Message*) 179
Can't determine IMAGETYPE for {LPT} (*Error*
 Message) 178

Cancel (*Option*) 82,88
Card (*Menu*) 67
Card parameters 140
Card types
 Browser 112
 Document 109
 LinkIndex 108
 Search 107
Card Types (*Menu*) 39,84
Card Types Menu Option 107
Cards 38
 closing 39
 creating 39
 creating new 83
 FileBox 92
 Graph 94
 graph-based 73
 lost 68
 opening 37
 parts 193
 size 42
 Sketch 93
 Sketch-based 72,100
 temporary Filebox 83
 Text 91
 text-based 72, 99
 types 63
 unfiled 68
 unspecified Filebox 83
 user 91
Cards and Boxes (*Option*) 111
Cards only (*Option*) 111
Case sensitivity 1,20
Certainty Value property 85
Change Card Title (*Option*) 75
Change Display Mode (*Option*) 75
Change Label (*Option*) 97
Change Link Type (*Option*) 75
Change Num (*Option*) 82
Change Scale (*Option*) 101
CHANGE.DEFAULT (*Option*) 170
ChangeBrowserRoots (*Option*) 116
ChangeBrowserSpecs (*Option*) 116
ChangeOverviewSpecs (*Option*) 116
Characters, special 20
Checkpoint (*Option*) 125
Checkpoint (*Menu*) 126
Checkpoint All (*Option*) 126
Checkpointing 195
Choose Default Printer (*Menu*) 174
Choose Display (*Option*) 56
Choose Font Profile to Update (*Menu*) 58
Choose Printer to Remove (*Menu*) 175
Choose Shade (*Menu*) 48
Clear (*Option*) 49,105
CLEAR how much? (*Menu*) 105
Clocks
 analog 167
 digital 163
Clocks (*Menu*) 54
Clocks (*Option*) 54
Close (*Menu*) 127
Close (*Option*) 46,127
Close All (*Option*) 127
Close All w/o confirm (*Option*) 127
Close and Close NoteCards (*Option*) 129

Close and Close Notefile (Option) 127
Close and Save (Option) 39,89
Close and Save (Option) 89
Close Cards Off Screen (Parameter) 140
Close Structure (Option) 129
Close w/o confirm (Option) 127
Close w/o Saving (Option) 39,89
Closing
 cards 39
 notefiles 35
Compact (Menu) 123
Compact (Option) 123
Compact In Place (Option) 124
Compact To New File (Option) 123
Compatibility
 file 18
 sysout 18
Compile (Option) 158
Configuring, software 11
Connect Nodes (Option) 96
Console messages 18
Conventions
 common {DSK} and {UNIX} 20
 file naming 19
 fonts 2
 keyboard 3
 notation 2
 NoteCards devices 2
 prompts 3
 {DSK} naming 20
 {UNIX} naming 22
Copy (Menu) 131
Copy (Option) 124,130,153
Copy protection 10
CopyEmbeddedLinks (Option) 111
Create (Menu) 125
Create (Option) 125
Create (Menu) 34
Create and Open (Option) 34,125
Create without Open (Option) 34,125
Creating cards 39
CREATIONDATE (File Attribute) 25
Cross File Link Mode (Parameter) 143
Cross-file links 77
 appearance 77
 missing notefiles 77
Cursor (Option) 106

D
Damaged Notefile 122
Dashed Links (Option) 114
Data Area 193
Default Card Type (Parameter) 140
Default Font (Option) 60
Default Font (Parameter) 138
Default printer
 adding 173
 removing 175
 setting 174
DEFAULTOSTYPE (Variable) 238
DEFAULTPRINTERTYPE (Variable) 50,239
DEFAULTPRINTINGHOST (Variable) 238
Del TEdit Process When Shrinking (Parameter) 137
Delete (Menu) 152
Delete (Option) 124,130
Delete Alarm Setting (Option) 166

Delete Card (Option) 89
Delete Link (Option) 97
Delete Node (Option) 96
Delete Old Versions (Option) 152
Delete Selected Property (Option) 86
Delete This Window (Option) 166
Designate FileBoxes (Option) 87
Digital clock
 12-hour/24 hour toggle 166
 adding new time zone 166
 change time zone 166
 changing 164
 deleting alarm setting 166
 setting AUX clock font 167
 setting AUX clock in current window 167
 setting alarm 165
 setting alarm loudness 166
 setting font 164
 setting regional time zone 167
 setting time 165
 setting time zone heading 167
 sizing display 166
 starting 163
 stopping 163
Directed/Undirected (Option) 98
DIRECTORIES (Variable) 236
Directory
 creating 24
 deletion 24
 enumeration 23
 home 20
 parent 20
 root 20
 working 20
Directory connector 170
 changing fonts 170
 starting 170
 stopping 170
 using 171
Directory Connector (Option) 55
Disconnect Nodes (Option) 97
Disk space 5
DISPLAYFONTDIRECTORIES (Variable) 237
Do what with deleted files? (Menu) 150
Document cards 84,109
Documentation kit 6
Don't Expunge (Option) 150
Done (Option) 88
{DSK}
 naming conventions 20
 special characters 20
{DSK}INIT.LISP (Variable) 235

E
Edit (Menu) 158
Edit (Option) 157
Edit Parameters (Option) 131,135
Edit Property List (Option) 85
Emulator 1
End-of-line convention 18
\\EndDST (Variable) 240
Error messages 187
Error recovery
 URAIID debugger 187,188

Errors

- break window 32
- correcting 32
- fatal 189
- file system 25
- I/O 187
- NoteCards 187
- system 189
- virtual memory 188

Exiting NoteCards 36

Expand (Option) 52,53

Expand on Bottom (Option) 102

Expand on Left (Option) 102

Expand on Right (Option) 102

Expand on Top (Option) 102

ExpandBrowserNode (Option) 115

ExpandEmbeddedLinks (Option) 111

Expunge (Option) 158

Expunge deleted files (Option) 150

Extra TEdit Props (Parameter) 136

F

Fast SEE Pretty (Option) 155

Fast SEE Unformatted (Option) 155

fg (UNIX Command) 58

File attributes 25

File capacity 81

File compatibility 18

File name conventions 19

File system errors 25

File type? (Menu) 50

FileBox

- card parameters 141

- cards 92

- Orphans 83

- Table of Contents 83

- To be Filed 83

FileBox Card (Menu) 90,93

FileBoxes 84

- accessing 206

- creating 203

- general 92

- system parameters affecting 93

Filebrowse (Option) 155,157

FileBrowser 145

- aborting commands 149

- browser window 147

- command menu 147

- command requiring input 149

- commands 151

- copy-selecting files 151

- copying files 153

- exiting 150

- hardcopy directory listings 151

- hardcopying 154

- info menu 148

- prompt window 147

- removing files 151

- renaming files 154

- scroll bar 148

- seeing (reviewing) file 155

- selecting files 148

- specifying browse files 145

- starting 145

- tally window 147

- troubleshooting 162

FileBrowser (Option) 54

FileBrowser Display Font (Option) 59

FileBrowser Prompt Font (Option) 59

Files, open 24

FIX MENU (Option) 99

Flush image & Exit (Option) 36,58

Font directories 7,175

- adding 175

- removing 177

- setting 175

- viewing 176

Font not Found (Error Message) 179

Fonts 2

- conventions 2

- default 60

- FileBrowser display 59

- FileBrowser prompt 59

- menu 60

- nonexistent 59

- parameters 137

- prompt window 59

- selecting display 58

Force Filing (Parameter) 140

Force Titles (Parameter) 140

Forget (Option) 56

From Contents (Option) 124

From Links (Option) 124

G

Ghost frame 51

Global default (Option) 161

Global link, adding 72

Graph cards 84,94

Graph-based cards 73

- deleting links from contents 74

Graph-Card (Menu) 94

GraphEditMenu (Option) 115

Grapher (Menu) 94

Grid On/Off (Option) 105

GridSize (Option) 105

Group (Option) 57

H

Hand Edit (Option) 101

HANDS (Option) 169

Hardcopy (Menu) 49,155

Hardcopy (Option) 49,54

HeadingsFromFileBoxes (Option) 110

Host access key

- obtaining 10

- using 12

I

I/O Errors 187

Icons

- shrink

Icon (Menu) 52

Icons, shrink 31

- FileBrowser window 52

- MenuBox 52

- Notecard 52

- Sketch window 52

- TEdit window 52

Idle (Menu) 55

Idle (Option) 55

Idle Display (Menu) 56

IDLE.PROFILE (Variable) 239
Include Card Object In ShowInfo (Parameter) 137
Include cards at: (Menu) 130
Index 193
Indicate NoteFile (Option) 87
Indicators Off (Option) 131
Infinite (Option) 161
Input/output devices, requirements 5,6
Insert Link (Option) 69,88
Insert Links (Option) 71
Inspect & Repair (Menu) 124
Inspect & Repair (Option) 123,124
Installation, software 9
InterPress, printing 154
INTERPRESSFONTDIRECTORIES (Variable) 237
Interrupt Which Process? (Menu) 185

K

Keyboard 15
 conventions 3
 Sun 3 16
 Sun 4 17

L

label larger (Option) 97
label smaller (Option) 97
Landscape
 description 179
 printer setting 179
Landscaped sketch (Option) 54
Larger Label (Option) 97
Ide 11
Ideether 11
LDEINIT (Variable) 235
LDRESOURCESYSOUT (Variable) 14
Left angle bracket 20
Link categories 63
 from-links 63
 global to-links 63
 to-links 63
Link Dashing In Browser (Parameter) 142
Link Icon Attach Bitmap Default (Parameter) 144
Link Icon Border Width (Parameter) 142
Link Icon Font (Parameter) 138
Link Icon Max Width in Pixels (Parameter) 143
Link Icon Multi Line Mode (Parameter) 143
Link Icon Show Link Type Default (Parameter) 144
Link Icon Show Title Default (Parameter) 144
Link icons
 active regions 64
 functionality 64
 mouse button use with 65
Link Index Specs? (Menu) 109
Link labels
 accessing 213
 creating 213
Link Ops (Menu) 75
Link parameters 142
Link type 42
Link Type (Menu) 69
Link types 42
 BrowserContents 61
 DocBack 62
 FiledCard 61
 LinkIndexBackPtr 62

 ListContents 61
 SubBox 61
 unspecified 42
 user-specified
 Comment 62
 See 62
 Unspecified 62
 customizing 62
 vs. card types 63
LinkIndex cards 84,108
Links 72
 access and use 64
 accessing 211
 adding global 72
 adding multiple global 72
 back 63
 backward 63
 creating 69,211
 creating in text cards 69
 cross-file 77
 deleting from card contents 73
 deleting from show links display 74
 directions 63
 ends 63
 forward 63
 global 63
 global, viewing 67
 inserting multiple 71
 inserting single 69
 local 63
 local, viewing 67
 storing 196
 system parameters for 76
 system-reserved 61
 tailoring 75
 traversing 37
 types 61
 user-specified 62
LISPUSERSDIRECTORIES (Variable) 236
Load (Option) 158
Load Patches (Option) 133
Locked (Option) 57
Logout (Option) 57
Logout / Save image & Exit (Option) 36
LONG-SITE-NAME (Variable) 240
Low Level Tools (Option) 58

M

Make Document Parameters (Menu) 110
Make this the new default? (Menu) 50
Markers In FileBoxes (Parameter) 141
Memory
 requirement 5
Menu Font (Option) 60
Menu Font (Parameter) 138
Menu Lingers After Notefile Close (Parameter) 140
MenuBox Icon 121
Mode (Menu) 47
Mouse
 buttons
 left 28
 miffir 28
 right 28
 using 27
 new, spawning 186
 screen sensitivity to 28

Move (Menu) 131
Move (Option) 51,131
Move Node (Option) 95
Move Node & SubTree (Option) 96
Move Region (Option) 96
Move Single Node (Option) 96
Multiple Roots (Option) 114
Multiple users 42

N

Naming conventions
 NoteCards devices 2
 common {DSK} and {UNIX} 20
 file 19
 {DSK} 20
 {UNIX} 22
NC FileBrowser (Option) 128
New cards 83
New Cards (Option) 39
New Info (Option) 160
New Notefile Initial Size (Parameter) 140
New Pattern (Option) 159
New Process Limit (Option) 133
New Wait Time (Option) 133
NewCards (Option) 83
NF Indicators On (Menu) 132
NF Indicators On (Option) 131
NO.HANDS (Option) 169
NO.NUMBERS (Option) 169
NO.RINGS (Option) 169
NO.TIMES (Option) 169
NONE (Option) 110,111
Notation conventions 2
NotBold (Option) 110
NoteCard types
 accessing 199
 creating 199
NoteCards 13
 UIDs 193
 accessing 206
 creating 203
 documentation kit 6
 exiting 14,35,36
 interface, customizing 214
 remote running 18
 specializing 43
 starting 32
 structure 41
 system parameters 136
 unique identifiers 193
NoteCards (Option) 54
NoteCards Parameters (Menu) 135
Notefile
 accessing 197
 banner 79
 checkpointing 195
 closing 35
 Abort 35
 Close 35
 creating 197
 creating new 34
 header 193
 index
 rebuild using notefile links plus 124
 rebuild using only notefile links 124
 indicators 132
 opening 32

 parameters 139
 structure 193
notefile 42
Notefile Inspector 225
Notefile Ops (Menu) 79,121
Notefile parameters 139
Notefiles 2
 damaged 122
 repairing inconsistent links 196
 saving contents 125
Number (Menu) 105
Number Pad (Menu) 55
NumberedBold (Option) 110
NUMBERS (Option) 169

O

OK (Option) 106
Open (Option) 122
Open file limit 24
Open Read-only (Option) 122
Open Read/Write (Option) 122
Operating system
 requirements 6
Operations on Bitmaps (Menu) 100
Orphans FileBox 83
Other Ops (Menu) 135

P

Page sized sketch (Option) 53
Paint (Option) 47,104
Paint Command (Menu) 47
Passwords
 authenticating 57
Pathname
 default 24
Pathnames 22
Period
 double 20,24
 single 20,24
Please select a font: (Menu) 59
Please select a font: (Menu) 137
POINTS (Option) 169
Portrait
 description 179
 printer setting 179
PostScript
 long edge points 182
 long edge shift 182
 printing 154
PostScript Font Association (Menu) 180
Postscript Parameters (Option) 58
PostScript Printer
 customizing 179
PostScript Short Edge Points 181
POSTSCRIPTFONTDIRECTORIES (Variable) 238
Print
 InterPress files 154
 PostScript files 154
 Sketch files 154
 TEdit files 154
 text files 154
Printer
 adding default 173
 default, removing 175
 default, setting 174
 setting up 173

troubleshooting 178
 Processor hardware, requirements 5
Prompt Window Font (Option) 59
 PUP protocol
 enabling 11
 PUP Ethernet 14
Put Cards Here (Option) 90

Q

Quiet Alarm/Loud Alarm (Option) 166
Quit - Saving Changes (Option) 86
Quit w/o Saving Changes (Option) 86

R

Recompute (Menu) 159
Recompute (Option) 159
Recompute Browser (Option) 113
RecomputeBrowser (Option) 115
ReconnectNodes (Option) 115
Redisplay (Option) 49
Reflect Diagonally (Option) 101
Reflect Left-to-right (Option) 101
Reflect Top-to-bottom (Option) 101
RelayoutGraph (Option) 115
 Release contents 6
 Relinking 11
Remove Default Printer (Option) 175
Remove Font Directory (Option) 177
Remove from which directory group? (Menu) 177
Remove Node (Option) 96
Remove which directory? (Menu) 177
Rename (Option) 124,154
 Requirements
 disk space 5
 hardware 5
 input/output devices 5
 memory 5
 operating system 6
 swap space 5
 tape access 6
Reset (Option) 105
RESET how much? (Menu) 105
RINGS (Option) 169
rmdir (UNIX Command) 24
Rotate Left (Option) 101
Rotate Right (Option) 101

S

Same Pattern (Option) 159
Save image & Exit (Option) 58
Save VM (Option) 58
Save w/o Closing (Option) 89
 Saving State 14
 Scroll
 bars 31
 down 31
 jump 31
 up 31
Scrollable & Pretty (Option) 155
Search Card (Menu) 107
 Search Cards 107
See (Menu) 156
See (Option) 155
See Window Control (Menu) 156
Set Alarm (Option) 165

Set Aux Clock Font In Just This Window (Option) 167
Set Default Display Font (Option) 58
Set Default Printer (Option) 58,174
Set Depth (Menu) 161
Set Depth (Option) 160,161
Set Font (Option) 164
Set Font for Aux Clocks (Option) 167
Set Local Time Zone (Option) 166
Set Regional Time Zone (Option) 167
Set Shape (Option) 48
Set Size (Option) 49
Set Time (Option) 165
Set Time-Zone Heading (Option) 167
Set Timeout (Option) 55
SET.TO.DEFAULT (Option) 170
SetMode (Option) 47
SetShade (Option) 48
SETTIME (Option) 170
Shape (Menu) 48
Shape (Option) 51
Shape to Fit (Option) 160,166
SHORT-SITE-NAME (Variable) 240
Show Font Directories (Option) 176
Show Info (Option) 86
Show Links (Option) 63,86
Show Links (Option) 67
Show Notefile On Cards (Parameter) 139
Show Profile (Option) 55
Show which directory group? (Menu) 177
SHOW.STYLE (Option) 169
ShowAsTile (Option) 105
Shrink (Option) 31,51
 Shrink icon 31
 expanding 31
 moving 31
 see also *Icons, shrink*
 Shrinking cards 38
Sides/Centers (Option) 99
Single-Item Snap (Menu) 100
 Site Initialization File 13,235
 locating 235
 structure 236
 Site variables 236
 LONG-SITE-NAME 240
 SHORT-SITE-NAME 240
 DEFAULTOSTYPE 238
 DEFAULTPRINTERTYPE 239
 DEFAULTPRINTINGHOST 238
 DIRECTORIES 237
 DISPLAYFONTDIRECTORIES 237
 IDLE.PROFILE 239
 INTERPRESSFONTDIRECTORIES 237
 LISPUSERSDIRECTORIES 236
 POSTSCRIPTFONTDIRECTORIES 238
 USERGREETFILES 238
 \\BeginDST 240
 \\EndDST 240
SIZE (File Attribute) 25
Size (Menu) 49
 Sketch
 printing 154
Sketch (Option) 53
 Sketch cards 84,85,93
 parameters 142
 system parameters affecting 94
Sketch, from a file (Option) 54
 Sketch-based cards 72,85,100

deleting links from contents 74
Sketch-Card (Menu) 93
 Slash 20
Smaller Label (Option) 97
Snap (Option) 46,54
Sort (Option) 162
Special Browser Specs (Option) 141
Special Cards (Option) 82
 Special characters 20
 Specializing NoteCards 43
 Starting NoteCards 13,32
STOP (Option) 99
 Structure
 Structures
 almost flat system 42
 flat system 42
 non-standard 41
 standard 41
 types 42
 SunOS
 directory notations 20
 files from NoteCards 19
Suspend & Exit (Option) 36,58
 Swap space
 adding 9
 ensuring adequate 9
 requirement 5
Switch Black & White (Option) 102
 Sysout 2
 compatibility 18
 location of 14
 System cards 84
 System parameters
 accessing 135
 browser card 141
 card 140
 filebox card 141
 font 137
 link 142
 NoteCards 136
 notefile 139
 sketch card 142
 System status 185

T
 Table of Contents FileBox 83
 TEdit 132
 printing 154
TEdit (Option) 53
TEdit Killer Off (Option) 133
TEdit Killer On (Menu) 132
TEdit Killer On (Option) 132
 Text cards 84,91
 Text-based cards 72,99
 deleting links from contents 73
Text-Card (Menu) 91
 Texture bitmap editor 102
 tilde 20
TIMES (Option) 169
Title/FileBoxes (Option) 88
TitlesFromNoteCards (Option) 110
To a file (Option) 155
To a printer (Option) 155
 To Be Filed FileBox 83
Toggle Border (Option) 97
Toggle Shade (Option) 97
Trim (Option) 101

Truncate File (Option) 122
TYPE (File Attribute) 25

U

UIDs 193
UnconnectNodes (Option) 115
Undelete (Menu) 152
Undelete ALL Files (Option) 153
Undo (Option) 88
Unfile from FileBoxes (Option) 88
 {UNIX}
 naming conventions 20,22
Unlocked (Option) 57
UnnumberedBold (Option) 110
 URAID 188
 commands 188
 entering 188
Use Deleted Link Icon Indicators (Parameter) 144
 User cards 84,91
USERGREETFILES (Variable) 238

V

Version numbering 21
 Virtual memory
 errors 188
 saving 15

W

Which printer? (Menu) 50
 Wild card characters
 * 107
 ? 107
Window Menu (Menu) 45
 Windows and the Window Title Bar 29
WRITEDATE (File Attribute) 25

X

XNS 14
 XNS Ethernet protocol
 enabling 11

Y

Yes (Option) 82

\

\\BeginDST (Variable) 240
\\EndDST (Variable) 240

~

~ 20
~/lisp.virtualmem (Variable) 14

*

LONG-SITE-NAME (Variable) 240
SHORT-SITE-NAME (Variable) 240

.

. 20
 .. 20

1

12-Hour Clock/24-Hour Clock (*Option*) 166

4

4x4 Shade Tool 48

<

<-> Sides (*Option*) 99

<.> Border (*Option*) 97

<.> Directed (*Option*) 98

<.> Shade (*Option*) 97

A

Abort (Option) 7,16
Abort All (Option) 8
Abort and Abort Notefile (Option) 8
 Abort current process 5
 Aborting Operations 1
Add Border (Option) 12
Add Global Link (Option) 12
Add Label (Option) 7
Add Link (Option) 6
Add New Property (Option) 8
Add New Regional Time Zone (Option) 4
Add Node (Option) 7
Add to which directory group? (Menu) 4
Allowed Logins (Menu) 12
Allowed Logins (Option) 12
Alphabetized FileBox Children (Parameter) 7
 Analog clock
 starting 6
Any Login (Option) 13
Arrow Heads In Browsers (Parameter) 8
Arrowheads (Option) 8
Assign Title (Option) 10
Attach Sketch Menu (Parameter) 8
Authenticate (Option) 13

B

Back links 3,5
Background (Menu) 9
 Backslash 8
 Banner
 notefile 1
\\BeginDST (Variable) 6
 Bitmap
 texture editor 12
Bitmap Editor (Menu) 14
 Bitmap operations 10
Bold (Option) 4
Bouncing Box (Option) 12
Bouncing Username (Option) 12
Boxes only (Option) 5
 Break window 6
 Break Windows 2
Bring Up Card/Box (Option) 15
Bring Up Cards At Previous Pos (Parameter) 7
 Browser Card
 parameters 7
 Browser cards 6
Browser Specs (Option) 8
Browser Specs? (Menu) 7
BrowserOverviewWin (Option) 10
BuildBacklinks (Option) 5
Bury (Option) 5

C

C-Shell 8

Can't Convert a POSTSCRIPT for a NIL Printer

(Error Message) 7

Can't determine IMAGETYPE for {LPT} (Error

Message) 6

Cancel (Option) 4,10

Card (Menu) 7

Card parameters 6

Card TYPES

 Search 1,2,3,6

Card Types (Menu) 6,13

Card Types Menu Option 1

Cards 12

 opening 1,2,3,4,5,8,9,10,11,12,13

Cards and Boxes (Option) 5

Cards only (Option) 5

Case sensitivity 1,8

Certainty Value property 7

Change Card Title (Option) 15

Change Display Mode (Option) 15

Change Label (Option) 7

Change Link Type (Option) 15

Change Num (Option) 4

Change Scale (Option) 11

CHANGE.DEFAULT (Option) 8

ChangeBrowserRoots (Option) 10

ChangeBrowserSpecs (Option) 10

ChangeOverviewSpecs (Option) 10

Characters, special 8

Checkpoint (Option) 5

Checkpoint (Menu) 6

Checkpoint All (Option) 6

Checkpointing 3

Choose Default Printer (Menu) 2

Choose Display (Option) 12

Choose Font Profile to Update (Menu) 14

Choose Printer to Remove (Menu) 3

Choose Shade (Menu) 4

Clear (Option) 5,15

CLEAR how much? (Menu) 15

Clocks

 digital 1,5

Clocks (Menu) 10

Clocks (Option) 10

Close (Menu) 7

Close (Option) 2,7

Close All (Option) 7

Close All w/o confirm (Option) 7

Close and Close NoteCards (Option) 9

Close and Close Notefile (Option) 7

Close and Save (Option) 11,13

Close and Save (Option) 11

Close Cards Off Screen (Parameter) 6

Close Structure (Option) 9

Close w/o confirm (Option) 7

Close w/o Saving (Option) 11,13

Closing

 notefiles 9,13

Compact (Menu) 3

Compact (Option) 3

Compact In Place (Option) 4
Compact To New File (Option) 3
Compile (Option) 14
Configuring
 software 3
Connect Nodes (Option) 6
Console messages 6
Conventions
 NoteCards devices 2,3,8,10
Copy (Menu) 11
Copy (Option) 4,9,10
Copy protection 2
CopyEmbeddedLinks (Option) 5
Create (Menu) 5
Create (Option) 5
Create (Menu) 8
Create and Open (Option) 5,8
Create without Open (Option) 5,8
Creating cards 13
CREATIONDATE (File Attribute) 13
Cross File Link Mode (Parameter) 9
Cross-file links 17
 appearance 17
Cursor (Option) 16

D
Damaged Notefile 2
Dashed Links (Option) 8
Data Area 1
Default Card Type (Parameter) 6
Default Font (Option) 16
Default Font (Parameter) 4
Default printer
 adding 1,2,3
DEFAULTOSTYPE (Variable) 4
DEFAULTPRINTERTYPE (Variable) 5,6
DEFAULTPRINTINGHOST (Variable) 4
Del TEdit Process When Shrinking (Parameter) 3
Delete (Menu) 8
Delete (Option) 4,10
Delete Alarm Setting (Option) 4
Delete Card (Option) 11
Delete Link (Option) 7
Delete Node (Option) 6
Delete Old Versions (Option) 8
Delete Selected Property (Option) 8
Delete This Window (Option) 4
Designate FileBoxes (Option) 9
Digital Clock
 starting 1,2
Directed/Undirected (Option) 8
DIRECTORIES (Variable) 2
Directory
 root 8,11,12
 home 8
 parent 8
 working 8
Directory connector 8
 starting 8,9
Directory Connector (Option) 11
Disconnect Nodes (Option) 7
Disk Space 1
DISPLAYFONTDIRECTORIES (Variable) 3
Do what with deleted files? (Menu) 6
Document cards 3,6
Documentation kit 2
Don't Expunge (Option) 6
Done (Option) 10

{DSK}
 special characters 8
{DSK}INIT.LISP (Variable) 1

E
Edit (Menu) 14
Edit (Option) 13
Edit Parameters (Option) 1,11
Edit Property List (Option) 7
Emulator 1
end-of-line convention 6
\\EndDST (Variable) 6
Error
 messages 3
Error Messages 3
Error recovery
 URAIID debugger 3,4
Errors
 file system 3,4,5,6,13
Exiting NoteCards 10
Expand (Option) 8,9
Expand on Bottom (Option) 12
Expand on Left (Option) 12
Expand on Right (Option) 12
Expand on Top (Option) 12
ExpandBrowserNode (Option) 9
ExpandEmbeddedLinks (Option) 5
Expunge (Option) 14
Expunge deleted files (Option) 6
Extra TEdit Props (Parameter) 3

F
Fast SEE Pretty (Option) 11
Fast SEE Unformatted (Option) 11
fg (UNIX Command) 14
File
 compatibility 6
File attributes 13
File capacity 3
File name
 conventions 7
File system errors 13
File type? (Menu) 6
FileBox
 Table of Contents 5
FileBox Card (Menu) 12
FileBox card parameters 7
FileBox cards 2
FileBox-Card (Menu) 3
FileBoxes 6
 general 2,3,7,10
Filebrowse (Option) 11,13
FileBrowser 1
 starting 1,3,4,5,6,7,9,10,11
FileBrowser (Option) 10
FileBrowser
 command menu 3,4,18
FileBrowser Display Font (Option) 15
FileBrowser Prompt Font (Option) 15
Files
 open 12
FIX MENU (Option) 9
Flush image & Exit (Option) 10,14
Font
 directories 3
Font Directory
 adding 3

Font Directory
 setting 3
Font not Found (*Error Message*) 7
 Fonts 2
 conventions 2,4,14,15,16
Force Filing (*Parameter*) 6
Force Titles (*Parameter*) 6
Forget (*Option*) 12
From Contents (*Option*) 4
From Links (*Option*) 4

G
 Ghost frame 7
Global default (*Option*) 17
 Global link
 adding 12
 Graph cards 4,6
 Graph-based cards 13
 deleting links from contents 14
Graph-Card (*Menu*) 4
GraphEditMenu (*Option*) 9
Grapher (*Menu*) 4
Grid On/Off (*Option*) 15
GridSize (*Option*) 15
Group (*Option*) 13

H
Hand Edit (*Option*) 11
HANDS (*Option*) 7
Hardcopy (*Menu*) 5,11
Hardcopy (*Option*) 5,10
HeadingsFromFileBoxes (*Option*) 4
 Host access key
 obtaining 2,4

I
 I/O Errors 3
 Icons
 shrink
 MenuBar 8
Icon (*Menu*) 8
 Icons 5
 shrink
 TEdit window 8
Idle (*Menu*) 11
Idle (*Option*) 11
Idle Display (*Menu*) 12
IDLE.PROFILE (*Variable*) 5
Include Card Object In ShowInfo (*Parameter*) 3
Include cards at: (*Menu*) 10
 Index 1
Indicate NoteFile (*Option*) 9
Indicators Off (*Option*) 11
Infinite (*Option*) 17
 Input/output devices, requirements 1,2
Insert Link (*Option*) 9,10
Insert Links (*Option*) 11
Inspect & Repair (*Menu*) 4
Inspect & Repair (*Option*) 3,4
 Installation
 software 1
 InterPress
 printing 10
INTERPRESSFONTDIRECTORIES (*Variable*) 3
Interrupt Which Process? (*Menu*) 1

K
 Keyboard 3
 conventions 3,5

L
label larger (*Option*) 7
label smaller (*Option*) 7
 Landscape
 description 7
Landscaped sketch (*Option*) 10
Larger Label (*Option*) 7
 Ide 3
 Ideether 3
LDEINIT (*Variable*) 1
 Left angle bracket 8
 Link Categories 3
 to-links 3
Link Dashing In Browser (*Parameter*) 8
Link Icon Attach Bitmap Default (*Parameter*) 10
Link Icon Border Width (*Parameter*) 8
Link Icon Font (*Parameter*) 4
Link Icon Max Width in Pixels (*Parameter*) 9
Link Icon Multi Line Mode (*Parameter*) 9
Link Icon Show Link Type Default (*Parameter*) 10
Link Icon Show Title Default (*Parameter*) 10
 Link icons
 functionality 4,5
Link Index Specs? (*Menu*) 3
 Link labels
 creating 17
Link Ops (*Menu*) 15
 Link parameters 8
 Link type 2
 unspecified 2
Link Type (*Menu*) 9
 Link types
 SubBox 1,2,3
 Comment 2
 LinkIndex cards 2,6
 Links 12
 traversing 1,2,3,4,7,9,11,12,13,14,15,16,17
LISPUSERSDIRECTORIES (*Variable*) 2
Load (*Option*) 14
Load Patches (*Option*) 13
Locked (*Option*) 13
Logout (*Option*) 13
Logout / Save image & Exit (*Option*) 10
LONG-SITE-NAME (*Variable*) 6
Low Level Tools (*Option*) 14

M
Make Document Parameters (*Menu*) 4
Make this the new default? (*Menu*) 6
Markers In FileBoxes (*Parameter*) 7
 Memory
 requirement 1
Menu Font (*Option*) 16
Menu Font (*Parameter*) 4
Menu Lingers After Notefile Close (*Parameter*) 6
 MenuBox Icon 1
Mode (*Menu*) 3
 Mouse
 screen sensitivity to 2
 using 1,2
Move (*Menu*) 11
Move (*Option*) 7,11
Move Node (*Option*) 5

Move Node & SubTree (Option) 6
Move Region (Option) 6
Move Single Node (Option) 6
Multiple Roots (Option) 8
Multiple users 2

N

Naming conventions
 NoteCards devices 2
NC FileBrowser (Option) 8
New cards 5
New Cards (Option) 13
New Info (Option) 16
New Notefile Initial Size (Parameter) 6
New Pattern (Option) 15
New Process Limit (Option) 13
New Wait Time (Option) 13
NewCards (Option) 5
NF Indicators On (Menu) 12
NF Indicators On (Option) 11
NO.HANDS (Option) 7
NO.NUMBERS (Option) 7
NO.RINGS (Option) 7
NO.TIMES (Option) 7
NONE (Option) 4,5
Notation conventions 2
NotBold (Option) 4
NoteCard types
 creating 3
NoteCards 1
 documentation kit 1,2,3,6,7,9,10,18
NoteCards (Option) 10
NoteCards Parameters (Menu) 1
Notefile
 opening 1,3,6,8,9,12
 Close 9
notefile 2
Notefile index
 rebuild using only notefile links 4
Notefile Inspector 1
Notefile Ops (Menu) 1
Notefile parameters 5
Notefiles 2
 damaged 2,4,5
Number (Menu) 15
Number Pad (Menu) 11
NumberedBold (Option) 4
NUMBERS (Option) 7

O

OK (Option) 16
Open (Option) 2
Open Read-only (Option) 2
Open Read/Write (Option) 2
Operating system
 requirements 2
Operations on Bitmaps (Menu) 10
Orphans FileBox 5
Other Ops (Menu) 1

P

Page sized sketch (Option) 9
Paint (Option) 3,14
Paint Command (Menu) 3
Passwords
 authenticating 13
Pathname

 default 12
Pathnames 10
Period
 single 8
 double 8
Please select a font: (Menu) 15
Please select a font: (Menu) 4
POINTS (Option) 7
Portrait
 printer setting 7
PostScript
 printing 10
PostScript Font Association (Menu) 8
Postscript Parameters (Option) 14
PostScript Printer
 customizing 7
PostScript Short Edge Points 9
POSTSCRIPTFONTDIRECTORIES (Variable) 4
Print
 text files 10
Printer
 setting up 1
 adding 1,2,3,6
Processor hardware
 requirements 1
Prompt Window Font (Option) 15
PUP protocol
 enabling 3
Put Cards Here (Option) 12

Q

Quiet Alarm/Loud Alarm (Option) 4
Quit - Saving Changes (Option) 8
Quit w/o Saving Changes (Option) 8

R

Recompute (Menu) 15
Recompute (Option) 15
Recompute Browser (Option) 7
RecomputeBrowser (Option) 9
ReconnectNodes (Option) 9
Redisplay (Option) 5
Reflect Diagonally (Option) 11
Reflect Left-to-right (Option) 11
Reflect Top-to-bottom (Option) 11
RelayoutGraph (Option) 9
Release contents 2
Relinking 3
Remove Default Printer (Option) 3
Remove Font Directory (Option) 5
Remove from which directory group? (Menu) 5
Remove Node (Option) 6
Remove which directory? (Menu) 5
Rename (Option) 4,10
Requirements
 hardware 1,2
Reset (Option) 15
RESET how much? (Menu) 15
RINGS (Option) 7
rmdir (UNIX Command) 12
Rotate Left (Option) 11
Rotate Right (Option) 11

S

Same Pattern (Option) 15
Save image & Exit (Option) 14
Save VM (Option) 14

Save w/o Closing (Option) 11
 Saving State 2
 Scroll
 up 5
 Scroll bars 5
Scrollable & Pretty (Option) 11
Search Card (Menu) 1
 Search Cards 1
See (Menu) 12
See (Option) 11
Set Alarm (Option) 3
Set Aux Clock Font In Just This Window (Option) 5
Set Default Display Font (Option) 14
Set Default Printer (Option) 2,14
Set Depth (Menu) 17
Set Depth (Option) 16,17
Set Font (Option) 2
Set Font for Aux Clocks (Option) 5
Set Local Time Zone (Option) 4
Set Regional Time Zone (Option) 5
Set Shape (Option) 4
Set Size (Option) 5
Set Time (Option) 3
Set Time-Zone Heading (Option) 5
Set Timeout (Option) 11
SET.TO.DEFAULT (Option) 8
SetMode (Option) 3
SetShade (Option) 4
SETTIME (Option) 8
Shape (Menu) 4
Shape (Option) 7
Shape to Fit (Option) 4,16
SHORT-SITE-NAME (Variable) 6
Show Font Directories (Option) 4
Show Info (Option) 8
Show Links (Option) 3,8
Show Links (Option) 7
Show Notefile On Cards (Parameter) 5
Show Profile (Option) 11
Show which directory group? (Menu) 5
SHOW.STYLE (Option) 7
ShowAsTile (Option) 15
Shrink (Option) 5,7
 Shrink Icon 5
 expanding 5
 Shrinking cards 12
Sides/Centers (Option) 9
Single-Item Snap (Menu) 10
 Site Initialization File 1
 locating 1,2
 Site variables 2
 LISPUSERSDIRECTORIES 2,3,4,5,6
SIZE (File Attribute) 13
Size (Menu) 5
 Sketch
 printing 10
Sketch (Option) 9
 Sketch card parameters 8
 Sketch cards 3,6,7
 system parameters affecting 4
Sketch, from a file (Option) 10
 Sketch-based cards 7,10,12
 deleting links from contents 14
Sketch-Card (Menu) 3
 Slash 8
Smaller Label (Option) 7
Snap (Option) 2,10

Sort (Option) 18
Special Browser Specs (Option) 7
Special Cards (Option) 4
 Special characters 8
 Specializing NoteCards 3
 Starting NoteCards 6
STOP (Option) 9
 Structure
 flat system 2
 Structure
 non-standard 1
 Structures
 standard 1
 SunOS
 directory notations 8
Suspend & Exit (Option) 10,14
 Swap space
 requirement 1
Switch Black & White (Option) 12
 Sysout 2
 location of 2,6
 System cards 6
 System parameters
 accessing 1,3,4,5,6,7
 System Status 1

T

Table of Contents FileBox 5
 TEdit 12
 printing 10
TEdit (Option) 9
TEdit Killer Off (Option) 13
TEdit Killer On (Menu) 12
TEdit Killer On (Option) 12
 Text cards 1,6
 Text-based cards 9,12
 deleting links from contents 13
Text-Card (Menu) 1
 Texture bitmap editor 12
 tilde 8
TIMES (Option) 7
Title/FileBoxes (Option) 10
TitlesFromNoteCards (Option) 4
To a file (Option) 11
To a printer (Option) 11
 To Be Filed FileBox 5
Toggle Border (Option) 7
Toggle Shade (Option) 7
Trim (Option) 11
Truncate File (Option) 2
TYPE (File Attribute) 13

U

UUIDs 1
UnconnectNodes (Option) 9
Undelete (Menu) 8
Undelete ALL Files (Option) 9
Undo (Option) 10
Unfile from FileBoxes (Option) 10
 {UNIX}
 naming conventions 8
Unlocked (Option) 13
UnnumberedBold (Option) 4
 URAID 4
 entering 4
Use Deleted Link Icon Indicators (Parameter) 10
 User cards 6

USERGREETFILES *(Variable)* 4

V

Version numbering 9

Virtual Memory
errors 4

W

Which printer? *(Menu)* 6

Wild card characters
* 1

Window Menu *(Menu)* 1

Windows and the Window Title Bar 3

WRITE DATE *(File Attribute)* 13

X

XNS Ethernet protocol
enabling 3

Y

Yes *(Option)* 4

\

BeginDST *(Variable)* 6

EndDST *(Variable)* 6

~

~ 8

Font DirectorY
viewing 4,5

Keyboard
Sun 3 4

Links
deleting 13

See Window Control *(Menu)* 12

System parameters
sketch card 8

*

LONG-SITE-NAME *(Variable)* 6

SHORT-SITE-NAME *(Variable)* 6

.

. 8

.. 8

1

12-Hour Clock/24-Hour Clock *(Option)* 4

4

4x4 Shade Tool 4

<

<-> **Sides** *(Option)* 9

<.> **Border** *(Option)* 7

<.> **Directed** *(Option)* 8

<.> **Shade** *(Option)* 7